

PATENT ABSTRACTS OF JAPAN

(11)Publication number : 08-249290
(43)Date of publication of application : 27.09.1996

(51)Int.Cl.

G06F 15/16
G06F 13/00
G06F 13/00

(21)Application number : 07-050678

(71)Applicant : HITACHI LTD
HITACHI TOHOKU SOFTWARE KK

(22)Date of filing : 10.03.1995

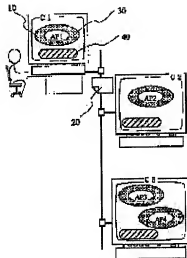
(72)Inventor : HICHI MASAHIRO
ITO TOSHIKI
OKAZAKI TSUKASA
AOKI YASUSHI
NAKANISHI JUN

(54) DISTRIBUTED SYSTEM

(57)Abstract:

PURPOSE: To provide a technique allowing an individual user to perform a cooperative processing between application programs proper to the utilizing method for himself by facilitating the change of the cooperative processing procedure between application programs.

CONSTITUTION: Plural Computers C1 to C3 are connected to this distributed system on a network. The system is provided with a service entity 10 obtained by giving a cooperative interface 30 cooperating with another application program in addition to application program main bodies AP1 to AP2 operating on a computer. Then, the system is provided with a service agent 20 which is provide with utilizing order information showing utilizing order information showing a moving destination between cooperative service entities and utilizing processing procedure information showing the procedure of processing to be used by each service entity 10 and moves on the network through the cooperative interface 30.



特開平8-249290

(43) 公開日 平成8年(1996)9月27日

| (51) Int.Cl. ⁴ | 識別記号 | 庁内整理番号 | F I | 技術表示箇所 |
|---------------------------|-------|---------|---------------|---------|
| G 0 6 F 15/16 | 3 7 0 | | G 0 6 F 15/16 | 3 7 0 N |
| 13/00 | 3 5 1 | 7368-5E | 13/00 | 3 5 1 H |
| | 3 5 5 | 7368-5E | | 3 5 5 |

審査請求 未請求 請求項の数 9 O L (全 33 頁)

(21) 出願番号 特願平7-50678

(22) 出願日 平成7年(1995)3月10日

(71) 出願人 000005108

株式会社日立製作所
東京都千代田区神田駿河台四丁目6番地

(71) 出願人 000233538

日立東北ソフトウェア株式会社
宮城県仙台市青葉区一番町2丁目4番1号

(72) 発明者 堀地 正浩

宮城県仙台市青葉区一番町二丁目4番1号
日立東北ソフトウェア株式会社内

(72) 発明者 伊藤 俊明

宮城県仙台市青葉区一番町二丁目4番1号
日立東北ソフトウェア株式会社内

(74) 代理人 弁理士 秋田 取喜

最終頁に続く

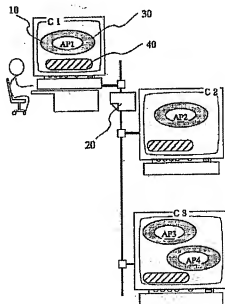
(54) 【発明の名称】 分散システム

(57) 【要約】

【目的】 アプリケーションプログラム間の連携処理手続きの変更を容易にし、個々の利用者が自分自身の利用方法に適するアプリケーションプログラム間の連携処理を行えることが可能な技術を提供すること。

【構成】 ネットワーク上に複数個のコンピュータが接続された分散システムであって、前記コンピュータ上で動作する各アプリケーションプログラム本体に他のアプリケーションプログラムと連携を行う連携インタフェースを付与してなるサービスエンティティを備え、連携させる該サービスエンティティ間の移動先を示す利用順序情報と各サービスエンティティの利用する処理の手続きを示した利用処理手続き情報とを有し、前記連携インタフェースを介してネットワーク上を移動するサービスエージェントを備える。

図 1



【特許請求の範囲】

【請求項1】 ネットワーク上に複数個のコンピュータが接続された分散システムであって、

前記コンピュータ上で動作する各アプリケーションプログラム本体に他のアプリケーションプログラムと連携を行う連携インタフェースを付与してなるサービスエンティティを備え、

連携させる該サービスエンティティ間の移動先を示す利用順序情報と各サービスエンティティの利用する処理の手続きを示した利用処理手続き情報とを有し、前記連携
10 インタフェースを介してネットワーク上を移動するサービスエージェントを備えたことを特徴とする分散システム。

【請求項2】 前記請求項1に記載の分散システムにおいて、

前記サービスエージェントの利用処理手続き情報は、サービスエージェントが利用する全てのサービスエンティティで使用するデータ項目の一覧を示す全データ項目、各サービスエンティティの入力データが格納されている入力データ項目、各サービスエンティティの処理
20 結果が格納される出力データ項目、入力データ項目に格納されているデータをサービスエンティティに入力する手続きを示すデータ入力手続き情報、出力データ項目ごとのデータの出力手続きを示すデータ出力手続き情報、データ出力手続きを用いて出力されたデータを処理する出力データ処理手続き情報、各サービスエンティティの中
のアプリケーションプログラムの実行を制御する制御手続き情報の少なくとも1つ以上を含むことを特徴とする分散システム。

【請求項3】 前記請求項2に記載の分散システムにおいて、

前記サービスエージェントの利用処理手続き情報の各構成要素をサービスエージェントを受け取ったサービスエンティティ、もしくはサービスエンティティの動作するコンピュータに応じて変更する手段を備えたことを特徴とする分散システム。

【請求項4】 請求項1記載の分散システムにおいて、前記サービスエンティティは、受信したサービスエージェントと同一の構成要素を持つ以上のサービスエージェントを複製する手段と、

複製したサービスエージェントを他の一つ以上のサービスエージェントに送信する手段と、

送信した1つ以上のサービスエージェントのいずれか1つのサービスエージェントがサービスエンティティに受け取られた時点で受け取り通知を他の複製されたサービスエージェントに通知する手段と、

受け取り通知を受け取ったサービスエージェントがサービスエージェント自身を削除する手段とを備えたことを特徴とする分散システム。

【請求項5】 前記請求項1に記載の分散システムにお

いて、

該コンピュータの各々は、該サービスエンティティ及び他のコンピュータとの間の通信路を確立、管理し、該通信路を通して受信した該サービスエージェントの有する利用順序に従ってサービスエージェントの移動を制御するルーティングマネージャを備えていることを特徴とする分散システム。

【請求項6】 請求項5に記載の分散システムにおいて、

前記ルーティングマネージャは、コンピュータ起動時に他のコンピュータと通信するための論理的な通信路を確立する論理的通信路確立手段と、

他のコンピュータとの間の該論理的通信路を切り放す手段と、

他のコンピュータからの接続要求を受け取る手段と、

ルーティングマネージャを停止させることなく、任意のコンピュータ間で論理的通信路の接続先コンピュータを
30 変更する手段と、

該ルーティングマネージャの動作するコンピュータ上で動作するサービスエンティティを管理する手段と、

受信したサービスエージェントの要求するサービスエンティティの名称を取得して、該ルーティングマネージャの動作するコンピュータ上で動作するサービスエンティティと該サービスエージェントが要求するサービスエンティティの名称を比較し、該コンピュータ上で該サービス
40 エージェントが要求するサービスエンティティが実行できるか否かを判定する手段と、

該サービスエージェントの要求するサービスエンティティが実行できない場合には、論理的に接続された他のコンピュータ上で動作しているルーティングマネージャに
50 該サービスエージェントを移動させる手段と、

該サービスエージェントの要求するサービスエンティティが実行できる場合には、該サービスエンティティに該サービスエージェントを移動させる手段とを備えたことを特徴とする分散システム。

【請求項7】 請求項6に記載の分散システムにおいて、前記論理的通信路確立手段は、前記論理的通信路を構成する接続ポートを設け、前記論理的通信路が構成されたと常に新たな接続ポートを生成する手段とを備えたことを特徴とする分散システム。

【請求項8】 前記請求項2に記載の分散システムにおいて、

前記サービスエージェントの構成要素である入力データ表示手続き情報は、入力データ項目と、グラフィカルユーザインタフェースの基本部品及びそれが操作された際の
60 処理基本部品を表示するグラフィカルユーザインタフェース基本部品表示手続きと入力データ項目を表示するグラフィカルユーザインタフェース部品の組み合わせ方とからなる組み合わせ手続きとから構成され、

前記入力データ項目と組み合わせ手続きとから、データ

3

に応じたグラフィカルユーザインタフェースを表示する手段を前記インタフェースに備えたことを特徴とする分散システム。

【請求項9】 前記請求項2に記載の分散システムにおいて、前記サービスエージェントの有する利用処理手続き情報の各手続き情報は、その手続きの処理内容を表す手続き名称と該手続き名称を持つ手続きのコンピュータネットワーク上の格納場所の組み合わせで記述され、

該手続き名称の手続きをその格納場所から読み込んで実行する手段を前記連携インタフェースに備えたことを特徴とする分散システム。

【発明の詳細な説明】

【0001】

【産業上の利用分野】 本発明は、コンピュータネットワークに接続されたコンピュータ上で動作する各種アプリケーションプログラムにより構成される分散システムに関し、特に、利用者が複数のアプリケーションプログラムを順次組み合わせながら利用して必要とする処理結果を得る業務や、個々の部署や作業者が各々々えられた作業を作業に必要なさまざまなアプリケーションプログラムを利用しながら進めていき、必要に応じてその処理経過や処理結果、処理内容を他の作業を行っている各作業者との間でやり取りすることにより、お互いの作業を連携させながら進めていく一連の作業から構成される業務に用いられる分散システムに適用して有効な技術に関するものである。

【0002】

【従来の技術】 分散システムは、個々の作業者が行うべき作業に用いられるコンピュータをネットワークで接続し、個々の作業者間で必要とするデータをやり取りするシステムである。

【0003】 このシステムでは、1つ1つの作業は個々の作業者がアプリケーションプログラムを用いて行う個人利用中心であり、各作業者の間で必要とするデータだけが共有され、各作業者の利用するアプリケーションプログラムで共有されたデータを利用できる。

【0004】 分散システム上でアプリケーションプログラムを連携して動作させるための仕組みに関しては、日経コンピュータ：クライアント・サーバーの弱点 非同期、蓄積型通信が解消、日経コンピュータ 1994. 8. 8. pp. 169-168.、「日経コンピュータ：分散アプリケーション同士の非同期連携を容易に実現するメッセージ・ベースのミドルウェアが急増」、日経コンピュータ 1993. 9. 20. pp. 67-74に、RPC (Remote Procedure Call)、会話型、ファイル転送、メッセージ・ベースの4つの連携方法について論じられている。

【0005】 RPC はアプリケーションプログラムの1つのモジュールを取りだし、そのモジュールを他のコンピュータで実行させる方法である。

【0006】 会話型は複数のアプリケーションプログラ

4

ムの間をリアルタイム通信路で接続し、そのリアルタイム通信路を通じてデータや処理要求をやり取りする方法である。この会話型の方法を用いて、ある作業に利用するアプリケーションプログラムを処理の要求元（クライアント）と処理の実行元（サーバ）に分割し、クライアント側アプリケーションプログラムとサーバ側アプリケーションプログラムの間で連携して処理を行えるようにしたもののがクライアント・サーバ・システムである。

【0007】 これらの2つの方法では、受信側アプリケーションプログラムは常に動作し、送信側アプリケーションプログラムからの処理要求を待つ状態にあり、送信側アプリケーションプログラムからの処理要求を受け取るとそれを即座に処理し、処理結果を送信側アプリケーションプログラムに返すことにより、送信側アプリケーションプログラムと受信側アプリケーションプログラムの間で同期を取りながら処理を行う連携処理を実現できる。

【0008】 また、これらの方法では、連携処理を行うアプリケーションプログラムの間の連携手続きをアプリケーションプログラムを開発する際にあらかじめ定め、その手続きをプログラムとして記述しておく必要がある。

【0009】 ファイル転送方法は、連携させるアプリケーションプログラムの間でファイルを転送し、ファイルに記述されたデータを受け取った場合にアプリケーションプログラムがそのデータに対する処理を行うものである。

【0010】 この方法では、ファイルの受信をトリガとしてアプリケーションプログラムを実行させることにより、非同期的なアプリケーションプログラム間の連携処理を実現できる。

【0011】 メッセージ・ベース方法は、送信側アプリケーションプログラムから送り出されたメッセージをメッセージ・ベースの処理を実現するプログラムが受け取り、メッセージに記述された受信側アプリケーションプログラムにそのメッセージを送り、メッセージを受け取った受信側アプリケーションプログラムではメッセージに対応した処理手続きを実行することにより、アプリケーションプログラム間の連携処理を実現する方法である。

【0012】 このメッセージ・ベース方法では、連携処理を行うアプリケーションプログラムの間であらかじめメッセージの形式を定め、そのメッセージを受け取った場合のアプリケーションプログラムの処理を記述するものである。

【0013】 これら4つのアプリケーションプログラム連携処理方法において、一連の作業から構成される業務を個々の作業で用いられる複数のアプリケーションプログラムの間で処理を連携させるためには、連携のために必要な処理手続きを各アプリケーションプログラムごと

5

に定め、プログラムとして記述しておく必要がある。

【0014】また、特開平2-186737号公報に開示されたプログラム間論理通信制御方式では、ネットワークに接続されたコンピュータ上で動作するアプリケーションプログラムの間で、データを交換することにより、アプリケーションプログラム間の連携処理を実現する方式について述べられている。

【0015】これは、アプリケーションプログラムの画面を介してユーザにより加えられた操作（データの変更、追加、削除など）のデータを論理的通信路を通じて同一のアプリケーションプログラム間でやり取りすることにより、そのアプリケーションプログラム間で画面の表示を共有することに基づくアプリケーションプログラムの間の連携処理を実現している。

【0016】これらのアプリケーションプログラム間の連携処理方法は、連携処理に用いるデータやメッセージを他のコンピュータ上のアプリケーションプログラムに送る際に、送信先のコンピュータの名称、送信先コンピュータの物理的な位置または物理的名称のいずれか1つの方法でデータやメッセージの送信先を指定することにより、指定されたコンピュータ上のアプリケーションプログラムにデータやメッセージを送ることができるものである。

【0017】さらに、最近では、個々の作業業者による個人利用中心のシステムから複数の作業業者の間にまたがる作業を支援するシステムの研究、開発が盛んに行われている。

【0018】これら複数の作業業者に跨る作業を支援するシステムの種類とその支援する作業の内容に関しては、石井裕：グループウェア技術の研究動向、情報処理 Vol. 30 No. 12 Dec. 1989、石井裕：コンピュータを用いたグループウェア支援の研究動向、コンピュータソフトウェア Vol. 8 No. 2 pp. 14-26、C. A. Ellis, S. J. Gibbs and G. L. Rein : "Groupware: Some Issues and Experiences" : CACM Jan. 1991 Vol. 34, No. 1 pp. 38-58、に国内外で研究、開発が行われている複数の人々の間に跨って行われる協同作業を支援するシステムが論じられている。

【0019】これらの協同作業を支援するシステムは、各作業業者が同一のアプリケーションプログラムを利用して同一の内容の作業を協同して行う（協同文書作成など）ことを支援するシステムであり、各作業業者の間でアプリケーションプログラムの処理結果を表示し、処理結果に対して何らかの操作を加えるユーザインタフェース画面を共有する機能を提供するものである。

【0020】

【発明が解決しようとする課題】本発明者は、上記従来技術を検討した結果、以下の問題点を見いだした。

【0021】上記、従来技術は、ある特定のアプリケーションプログラム間でやり取りするデータやメッセージ

6

をあらかじめ定め、その定められたデータやメッセージに対応する処理プログラムをそれぞれのアプリケーションプログラムに記述し、このあらかじめ定められたデータやメッセージをアプリケーションプログラム間でやり取りすることにより、アプリケーションプログラム間の連携処理を行うため、アプリケーションプログラム間の連携処理の手続きを変更することはできないか、変更できたとしても、変更に関連する全てのアプリケーションプログラム中の処理プログラムを変更する必要がある。

【0022】そのため、作業内容や目的に応じて容易にアプリケーションプログラム間の連携処理手続きを変更することは非常に困難となっており、アプリケーションプログラム間の連携処理手続きを変更するためには、その変更対象となる全てのアプリケーションプログラムの実行をいったん終了させる必要があった。

【0023】また、分散システム上ではさまざまな利用者が各々の作業を行うためにアプリケーションプログラムを使用しており、変更対象となる全てのアプリケーションプログラムの実行をいったん終了させることは、変更対象となるアプリケーションプログラムの数が増加するに従い、困難さを増してくることになる。

【0024】そして、アプリケーションプログラム間で連携処理を行うためにやり取りされるものは、データもしくはメッセージであるため、それらデータやメッセージを受け取った際の処理手続きは、それを受け取った受信側アプリケーションプログラムにより決められており、送信側アプリケーションプログラムが受信側アプリケーションプログラムの処理手続きを変更することは不可能であった。

【0025】さらに、受信側アプリケーションプログラムの提供する複数の機能を組み合わせた一連の処理手続き、すなわち受信側アプリケーションプログラムで一度に実行したい複数のコマンドの組み合わせの実行を実現するためには、毎回、回を繰り返さなければならない処理手続きを実行するためのメッセージをアプリケーションプログラムに送り、該当する手続きを実行し、その実行結果を送信側アプリケーションプログラムで組み合わせるか、あらかじめ受信側アプリケーションプログラムに複数のコマンドを実行するための特別なメッセージをプログラムとして記述しておく必要があった。

【0026】このように、あらかじめアプリケーションプログラム間の連携処理手続きを記述しておく必要があるため、個々の利用者が自分自身の利用方法に適するアプリケーションプログラム間の連携処理手続きを記述することは不可能であり、個々の利用者が自分自身の利用方法に適するアプリケーションプログラム間の連携処理を行うことはできなかったという問題点があった。

【0027】さらに、これらのデータやメッセージをアプリケーションプログラムに送るためには、そのアプリケーションプログラムの動作しているコンピュータをコ

ンピュータの名称、またはIP アドレス等のコンピュータを一意に識別するための手段で明示的に指定する必要があり、ある処理を行うアプリケーションプログラムの動作するコンピュータを何等かの理由で変更した場合に、該当するアプリケーションプログラムを利用しているアプリケーションプログラム連携処理手続きの中のコンピュータを指定している部分を変更しなければならず、変更が困難であるという問題点があった。

【0028】本発明の目的は、アプリケーションプログラム間の連携処理手続きの変更を容易にし、個々の利用者が自分自身の利用方法に適するアプリケーションプログラム間の連携処理を行えることが可能な技術を提供することにある。

【0029】本発明の前記ならびにその他の目的と新規な特徴は、本明細書の記述及び添付図面によって明らかなるであらう。

【0030】

【課題を解決するための手段】本願において開示される発明のうち、代表的なものとの概要を簡単に説明すれば、下記のとおりである。

【0031】ネットワーク上に複数個のコンピュータが接続された分散システムであって、アプリケーションプログラム間連携処理に関する個々のアプリケーションプログラム本体にアプリケーションプログラム間連携処理手続きを実行するための連携インタフェースを付したサービスエンティティと、連携させるサービスエンティティ間の移動先を表す利用順序とサービスエンティティの利用手続きを有するサービスエージェントと、サービスエージェントをそれらが有する利用順序に記述された移動先に移動させる通信路を生成、維持、管理するルーティングマネージャとを備える。

【0032】そのサービスエンティティは、受信したサービスエージェントと同一の構成要素を持つ一つ以上のサービスエージェントを複製する手段と、複製したサービスエージェントを他の一つ以上のサービスエージェントに送信する手段と、送信した一つ以上のサービスエージェントのいずれか一つ以上のサービスエージェントがサービスエンティティに受け取られた時点で受け取り通知を他の複製されたサービスエージェントに通知する手段と、受け取り通知を受け取ったサービスエージェントがサービスエンティティ自身を削除する手段とを備える。

【0033】そのサービスエージェントは、連携処理においてアプリケーションプログラムを連携させる順序を該当するアプリケーションプログラム本体を有するサービスエンティティの利用順序として記述した利用順序情報と、各サービスエンティティにおいて使用するアプリケーションプログラムの有する機能を利用する制御手続き情報、その制御手続き情報を用いてサービスエンティティの有するアプリケーションプログラムに実際に処理を行わせるために必要なデータを格納するデータ項目、

データ項目に格納されたデータをアプリケーションプログラムに渡し、その処理結果データを該当するデータ項目に格納する入出力データ処理手続き情報、サービスエージェントの有するデータをグラフィカルユーザインタフェースを通して利表示するめ表示手続き情報からなる利用手続き情報とを備える。

【0034】そのルーティングマネージャは、コンピュータ起動時に他のコンピュータと通信するための論理的な通信路を確立する論理的通信路確立手段と、他のコンピュータとの間の該論理的通信路を切り放す手段と、他のコンピュータからの接続要求を受け取る手段と、ルーティングマネージャを停止させることなく、任意のコンピュータ間で論理的通信路の接続先コンピュータを変更する手段と、該ルーティングマネージャの動作するコンピュータ上で動作するサービスエンティティを管理する手段と、受信したサービスエージェントの要求するサービスエンティティの名称を取得して、該ルーティングマネージャの動作するコンピュータ上で動作するサービスエンティティと該サービスエージェントが要求するサービスエンティティの名称を比較し、該コンピュータ上で該サービスエージェントが要求するサービスエンティティが実行できるか否かを判定する手段と、該サービスエージェントの要求するサービスエンティティが実行できない場合には、論理的に接続された他のコンピュータ上で動作しているルーティングマネージャに該サービスエージェントを移動させる手段と、該サービスエージェントの要求するサービスエンティティが実行できる場合には、該サービスエンティティに該サービスエージェントを移動させる手段とを備える。

【0035】

【作用】上述した手段によれば、分散システムを構成する各コンピュータ上で動作するルーティングマネージャは、サービスエージェントを送受信するための通信路を確立し、ルーティングマネージャの動作しているコンピュータ上で動作するサービスエンティティの名称とそのサービスエンティティとの間でサービスエージェントを送受信するための通信路を管理する。

【0036】サービスエンティティが起動されると、サービスエンティティの動作するコンピュータ上のルーティングマネージャとの間にサービスエージェントを送受信するための通信路を確立し、ルーティングマネージャにサービスエンティティの名称を送り、ルーティングマネージャは送られたサービスエンティティの名称とそのサービスエンティティへの通信路を管理する。

【0037】ルーティングマネージャは、他のコンピュータ上で動作しているルーティングマネージャから通信路を通して受け取ったサービスエージェントの利用順序に指定されたサービスエンティティがそれ自身の動作するコンピュータ上で利用できるか否かを判定し、利用できる場合にはサービスエージェントを該当するサービス

エンティティに送り、利用できない場合にはサービスエ
 ージェントを他のコンピュータ上のルーティングマネー
 ージャに移動させる。

【0038】サービスエンティティは、通信路を通して
 サービスエージェントを受け取ると、連携インタフェ
 ースがそのサービスエージェントの持つ利用手続きの中か
 らそのサービスエンティティで使用される手続きとデー
 タを取得し、取得した手続きを実行することにより、サ
 ービスエンティティの有するアプリケーションプログラム
 本体の処理を倒し、サービスエージェントの必要とし
 ている処理結果を得、その結果をサービスエージェン
 トに送り、サービスエージェント中のデータ項目に格納
 する。

【0039】サービスエージェントから取得した手続き
 に対応するプログラムが他のコンピュータ上にあること
 が指定されている場合には、連携インタフェースはその
 プログラムをサービスエンティティが動作しているコン
 ピュータ上にコピーし、そのプログラムを実行すること
 により、サービスエージェントの要求する手続きを実行し
 、サービスエージェントの必要としている処理結果を
 格納した後、サービスエンティティの連携インタフェ
 ースはサービスエージェントの移動先を利用順序にしたが
 って変更し、通信路を通じてルーティングマネージャに
 サービスエージェントを送る。

【0040】ルーティングマネージャは、サービスエン
 ティティから送られるサービスエージェントの利用順序
 に指定されたサービスエンティティがそれぞれ動作す
 るコンピュータ上で利用できるかを判定し、利用で
 きる場合にはサービスエージェントを該当するサービス
 エンティティに送り、利用できない場合にはサービスエ
 ージェントを他のコンピュータ上のルーティングマネー
 ージャに移動させる。

【0041】このように、アプリケーションプログラム
 間の連携処理手続きをサービスエージェントというア
 プリケーションプログラム本体とは独立した形式で与え
 、アプリケーションプログラム本体に連携処理のための連
 携インタフェースを付加し、この連携インタフェースが
 サービスエージェントの持つ利用手続きを実行すること
 によりアプリケーションプログラム間の連携処理をサ
 ービスエージェントを通して容易に変更することができ
 る。

【0042】また、業務に応じて、その業務を進めるた
 めに必要なさまざまなアプリケーションプログラムとそ
 の操作を組み合わせた利用手順、サービスエージェン
 トの中に一連の作業手順と各作業で使用されるアプリ
 ケーションプログラムの利用手続きとして記述し、このサ
 ービスエージェントをコンピュータネットワークに接続
 されたコンピュータ上で動作している各種サービスエン
 ティティの間で移動させ、サービスエージェントを受信
 したサービスエンティティはサービスエージェントに記

述された利用手続きを実行することにより、あらかじめ
 アプリケーションプログラム間の連携処理手続きを記述
 しておかなくても、個々の利用者が自分自身の利用方法
 に適するアプリケーションプログラム間の連携処理を行
 えることが可能となる。

【0043】以下、本発明について、実施例とともに説
 明する。

【0044】なお、実施例を説明するための図におい
 て、同一機能を有するものは同一符号を付け、その繰り
 返しの説明は省略する。

【0045】

【実施例】以下、本発明の一実施例である分散シス
 テムにおけるアプリケーションプログラム本体と連携イン
 タフェースから構成される複数のサービスエンティティ
 の間をサービスエージェントが移動しながら、サービスエ
 ンティティを通してアプリケーションプログラムを連携
 させ、処理を進めていくアプリケーションプログラム間
 連携処理について説明する。

【0046】図1は、本実施例の分散システムの構成を
 示した図である。

【0047】図1に示すように、本実施例の分散シス
 テムは、コンピュータC(C1~C3)、コンピュータ間
 でデータ通信を行うためのネットワーク0、各コンピ
 ュータ上で動作するアプリケーションプログラム本体A
 P(A P1~A P4)、他のアプリケーションプログラム
 と連携して処理を行うための連携インタフェース30、
 アプリケーションプログラム本体A Pに連携インタフェ
 ース30を付与したもので、アプリケーションプログラ
 ム間連携処理の基本となる処理単位であるサービスエ
 ンティティ10、各アプリケーションプログラム間を移動
 し連携処理を実現するサービスエージェント20、サー
 ビスエージェント20の移動する通信路及びサービスエ
 ンティティとの間の通信路を確立、管理し、サービスエ
 ージェント20をサービスエンティティ10間で移動さ
 せるルーティングマネージャ40から構成される。

【0048】本実施例の説明上、コンピュータCの各名
 称をそれぞれC1、C2、C3と記し、各アプリケーシ
 ョンプログラム本体A Pの各名称をそれぞれA P1、A
 P2、A P3、A P4と記し、各アプリケーションプロ
 グラム本体A P1~A P4を有するサービスエージェン
 トの名称をそれぞれSE1、SE2、SE3、SE4と
 記す。

【0049】本実施例の分散システムにおいて、コンピ
 ュータ1は、図2に示すように、キーボード11やマウ
 ス12などから構成される入力装置13、CPU14やメ
 モリ15を格納したコンピュータ本体である処理装置
 16、データを表示する表示装置17、データやアプリ
 ケーションプログラムを格納する外部記憶装置18、デ
 ータを印刷するプリンタ19から構成される。

【0050】入力装置13は、上記以外のタブレット、

11

タッチパネルでも良い。

【0051】また、入力装置13には、画像データを入力するためのスキャナ、音声を入力するためのマイクが加えられることがある。

【0052】入力装置13のマウスは、表示装置17上の位置を指定したり、表示装置17に表示されたいくつかの選択肢を含むメニューの中から選択対象を指定するための手段であり、このような指定手段を有する他の装置、例えば光学式ペンやタッチパネルであっても良い。

【0053】また、表示装置17には、音声を出力する

10

ためのスピーカが加えられることがある。

【0054】ネットワーク0は、ネットワークに接続されたコンピュータ間でデータを送受信する手段であり、特定の場所内のネットワークである LAN (Local Area Network)、各拠点間のネットワークである WAN (Wide Area Network)、ネットワーク同士を相互に接続したネットワークであるインターネットのいずれのネットワークであっても良い。

【0055】ルーティングマネージャ40は、各コンピュータごとに1つずつ存在し、コンピュータ起動時に起動される。

20

【0056】ルーティングマネージャ40をコンピュータ起動時に起動する方法としては、例えば、UNIX (A.T&Tの登録商標) のデーモンを用いる方法などがある。

【0057】本実施例では、上記のネットワーク0に接続されたコンピュータC1で作成されたサービスエージェント20がルーティングマネージャ40を通してネットワーク0に接続されたコンピュータC2、C3間を移動し、これらのコンピュータC2、C3上で動作するサービスエンティティ10の連携インタフェース30を通してその中のアプリケーションプログラム本体Aの処理を実行、制御し、アプリケーションプログラム間の連携処理を行う。

30

【0058】このようにサービスエージェント20を用いて複数のサービスエンティティ10を通してアプリケーションプログラム本体AのPの実行の制御、管理を行い、一連の作業に用いられるアプリケーションプログラム間を連携して処理することにより、一連の作業を自動化、支援する。

【0059】本発明による分散システムのアプリケーションプログラム間連携処理では、一台以上のコンピュータC上で動作する2つ以上のアプリケーションプログラム本体Aを有するサービスエンティティ10で利用することが可能であるが、本実施例では、図1に示すように、3台のコンピュータと各々のコンピュータ上で動作する4つのサービスエンティティ10をサービスエージェント20を用いて連携している。

【0060】上記以外の構成で利用する場合、コンピュータやアプリケーションプログラム本体Aを有するサ

ービスエンティティ10の数が増える点だけが本実施例と異なるのみである。

【0061】本実施例では、利用者がある作業を行う際に、アプリケーションプログラムA1→A2→A3もしくはA4をこの順序で連携して処理させる必要がある場合を取り挙げ、その各アプリケーションプログラムの処理を行うために、A1は項目1、項目2に格納されたデータを用いて処理を行い、その処理結果データを項目6に格納し、A2は項目3、項目4に格納されたデータを用いて処理を行い、その処理結果データを項目7に格納し、A3は項目4、項目6に格納されたデータを用いて処理を行い、その処理結果データを項目8に格納し、A4は項目5に格納されたデータを用いて処理を行い、その処理結果データを項目9に格納するものとする。なお、ここで示した項目1～項目9は、後述するサービスエージェントに示されたデータ項目である。

【0062】次に、本実施例のサービスエージェント20について説明する。

【0063】図3は、本実施例のサービスエージェント20の構造を説明するための図である。

【0064】図3に示すように、サービスエージェント20は、個々のサービスエージェント20を識別するための識別子201、利用順序情報 (サービスエージェントの移動先の一覧を記述した移動先リスト202、サービスエージェントが移動してきた経路を管理する移動リスト203、現時点でサービスエージェント20が必要としているアプリケーションプログラム本体Aを持つサービスエンティティ10の名称とそのサービスエンティティ10が動作しているコンピュータの名称を示す移動先名204) と、利用処理手続き情報 (連携処理を行う複数のサービスエンティティ10が必要とする全てのデータ項目 (211～219) を列挙した全データ項目21、全データ項目21の中の各データ項目 (211～219) に応じた出力方法を記述したデータ出力手続き22、全データ項目21の中の各データ項目やサービスエンティティ10に応じた入力方法を記述したデータ入力手続き23、データ出力手続き22を用いて入力されたデータを処理する方法である出力データ処理手続き24、サービスエージェント20の移動先を決定し、決定した移動先にサービスエージェント10を遷移するための方法である移動手続き25、各サービスエンティティ10の持つアプリケーションプログラム本体AのPの実行を制御するための手続きである制御手続き26) とから構成される。

【0065】サービスエージェント20の構造のうち、識別子201、移動先リスト202、移動リスト203、移動先名204は、サービスエージェント20の作成時に必ず作成される項目である。

【0066】上記以外の全データ項目21、データ出力

50

13

手続き22、データ入力手続き23、出力データ処理手続き24、移動手続き25、制御手続き26の各項目や手続きは、サービスエージェント20を用いて行われるサービスエンティティ10間の連携処理の方法ごとに異なる要素であり、全ての要素が必ず存在するわけではない。

【0067】全データ項目21は、何等かの値を持つデータ項目(211~216)である入力済みデータ項目27と、値を持たないデータ項目(217~219)である未入力データ項目28とから構成される。

【0068】入力済みデータ項目27は、サービスエージェント20があるサービスエンティティ10に受け取られた時に、そのサービスエンティティ10に渡すべきデータを格納しているデータ項目である入力データ項目271であり、未入力データ項目は、あるサービスエンティティ10の実行結果データを格納するデータ項目である出力データ項目281である。

【0069】例えば、図3において、アプリケーションプログラム本体AP1が実行時に必要とするデータが項目1(211)、項目2(212)に格納されており、その実行結果データが項目6(216)に格納されると、サービスエンティティSE1では、項目1(211)と項目2(212)が入力データ項目271、項目6(216)が出力データ項目281となる。

【0070】アプリケーションプログラム本体AP2が実行時に必要とするデータが項目3(213)、項目4(214)に格納されており、その実行結果データが項目7(217)に格納されると、サービスエンティティSE2では、項目3(213)と項目4(214)が入力データ項目271、項目7(217)が出力データ項目281となる。

【0071】アプリケーションプログラム本体AP3が実行時に必要とするデータが項目4(214)、項目6(216)に格納されており、その実行結果データが項目8(218)に格納されると、サービスエンティティSE3では、項目4(214)と項目6(216)が入力データ項目271、項目8(218)が出力データ項目281となる。

【0072】アプリケーションプログラム本体AP4が実行時に必要とするデータが項目5(215)に格納されており、その実行結果データが項目9(219)に格納されると、サービスエンティティSE4では、項目5(215)が入力データ項目271、項目9(219)が出力データ項目281となる。

【0073】このように入力データ項目271や出力データ項目281は、サービスエージェント20が受け取られたサービスエンティティ10ごとに異なる。

【0074】データ出力手続き22には、未入力データ項目28の中からサービスエンティティ10の持つアプリケーションプログラム本体APに応じた出力データ項

14

目281を選択するための手続きである出力データ選択手続き221、画面から入力された値を取得するための手続きである出力データ取得手続き222、アプリケーションプログラム本体APの出力結果データを取得するための手続きであるデータ取得手続き223、データ取得手続き223を用いて取得されたアプリケーションプログラム本体APの出力結果データを出力データ項目281のデータ形式に変換するための手続きである出力データ変換手続き224、出力データ取得手続き222を用いて画面から得られた値や出力データ変換手続き224を用いて出力データ項目281のデータ形式に変換された値を出力データ項目281に格納するための手続きである出力データ格納手続き225が含まれる。

【0075】データ入力手続き23には、入力済みデータ項目27の中からサービスエンティティ10の持つアプリケーションプログラム本体APに応じた入力データ項目271を選択するための手続きである入力データ選択手続き231、入力データ選択手続き231を用いて選択された入力データ項目271の値をアプリケーションプログラム本体APのデータ入力形式に変換するための手続きである入力データ変換手続き232、入力データ変換手続き232を用いて変換された入力データ項目271の値をアプリケーションプログラム本体APに渡すための手続きであるパラメータ設定手続き233、入力データ変換手続き232を用いて変換された入力データ項目271の値を画面に表示するための手続きである入力データ表示手続き234が含まれる。

【0076】出力データ処理手続き24は、出力データ取得手続き222やデータ取得手続き223を用いて取得されたデータを処理するための手続きである。

【0077】例えば、出力データ取得手続き222を用いて取得された複数のデータの和を計算する処理などがこの出力データ処理手続き24に記述される。

【0078】移動手続き25は、サービスエージェント20の移動先を決定し、決定した移動先にサービスエージェント20を送るための手続きである。

【0079】この移動手続き25の移動先の決定方法には、サービスエージェント作成時に与えられた移動順序に従って次の移動先を決定する方法、入力済みデータ項目27や未入力データ項目28の中の任意のデータ項目の組み合わせに応じてサービスエージェントの移動先を決定する方法が挙げられる。

【0080】制御手続き26は、サービスエンティティ10の連携インタフェース30で実行されるアプリケーションプログラム本体APの実行を制御するための手続きであり、アプリケーションプログラム本体APの起動・終了、アプリケーションプログラム本体APのコマンド実行、及び複数のコマンドの組み合わせである制御スクリプトがある。

【0081】次に、ルーティングマネージャ40につい

15

て説明する。

【0082】ルーティングマネージャ40は、複数のコンピュータ間の論理的通信路の確立と維持、各サービスエンティティ10との間の入出力チャネル403の確立、維持を行うものであり、図4～図6に各コンピュータ上で動作するルーティングマネージャ40により構築される論理的な通信路の接続形態例を示す。

【0083】論理的通信路は、相互に接続された2つのルーティングマネージャ40を接続する論理的通信路であるチャネル400の組み合わせから構成される。

【0084】図4に示す例では、コンピュータC1、C2、C3上で動作する3つのルーティングマネージャ10(RM1、RM2、RM3)の間で、RM1とRM2、RM2とRM3が接続され、RM1-RM2-RM3という論理的通信路を構成していることを表している。

【0085】図5に示す例では、図4に示した論理的通信路を構成しているルーティングマネージャ40(RM2)に新たなルーティングマネージャ40(RM4)が接続され、論理的通信路の構成が変化した状態を示した図である。

【0086】図6に示す例では、ある一台のコンピュータ内におけるルーティングマネージャ40(RM1)と二つのサービスエンティティ10(SE3、SE4)が入出力チャネル403により接続された接続状態、及びサービスエンティティ10を構成する連携インタフェース30とアプリケーションプログラム本体APの間がアプリケーションインタフェース38で接続された接続状態を表した図である。この図6では、図1に示したコンピュータC3とC2の接続の例を表している。

【0087】図6に示す入出力チャネル403は、ルーティングマネージャ40とサービスエンティティ10を接続した通信路であり、実際にはサービスエンティティ10内の連携インタフェース30とルーティングマネージャ40の間を接続する。

【0088】アプリケーションインタフェース38は、サービスエンティティ10内に生成される連携インタフェース30とアプリケーションプログラム本体APの入出力を接続するインタフェースである。

【0089】ルーティングマネージャ40は、チャネル400を接続するための受け口となるチャネルポート401を持ち、そのチャネルポートは、個々のチャネルポート401を識別するためのチャネルポート番号を持つ。チャネルポート番号は、ルーティングマネージャ40により、チャネルポート401生成時に与えられる。

【0090】すなわち、チャネル400とは2つのルーティングマネージャ40のチャネルポート401を接続した通信路であり、チャネル400を生成するとは、任意の2つのルーティングマネージャ40のチャネルポート401の間を接続することである。

16

【0091】ルーティングマネージャ40は、そのチャネル400を生成することによりチャネルポート401が使用されると新たな空きチャネルポート402を1つ生成する。

【0092】そのため、ルーティングマネージャ40は、常に他と接続されていない空きチャネルポート402を持つことになる。

【0093】この空きチャネルポート402は、他のルーティングマネージャ40から接続要求がなされた場合にそのルーティングマネージャ40との間でチャネル400を生成するために使用される。

【0094】ルーティングマネージャ40が生成した空きチャネルポート402は、サービスエンティティ10とルーティングマネージャ40を接続する際に、ルーティングマネージャ40同士を接続する場合と同様に用いられる。

【0095】すなわち、ルーティングマネージャ40とサービスエンティティ10の間に入出力チャネル403を生成する場合には、サービスエンティティ10が起動されたときに新たな空きチャネルポートをサービスエンティティ10の連携インタフェース30に生成し、サービスエンティティ10からルーティングマネージャ40に接続要求を行い、ルーティングマネージャ40が空きチャネルポート402とサービスエンティティ10の連携インタフェース30の空きチャネルポートの間に入出力チャネル403を生成し、ルーティングマネージャ40のみが新たな空きチャネルポート402を生成する。

【0096】ルーティングマネージャ40は、このようにチャネル400が生成されると、新たな空きチャネルポート402を1つ生成し、常に1つの空きチャネルポート402が残るようにチャネルポートの管理をする。

【0097】サービスエンティティ10は、起動時にルーティングマネージャ40との間で入出力チャネル403を生成するための空きチャネルポート402を連携インタフェース30に生成し、ルーティングマネージャ40との間で入出力チャネル403を生成することに加え、連携インタフェース30とアプリケーションプログラム本体APの入出力を接続するためのアプリケーションインタフェース38を持ち、このアプリケーションインタフェース38を用いて連携インタフェース30とアプリケーションプログラム本体APの間の入出力、すなわちコマンドを連携インタフェース30からアプリケーションプログラム本体APにコマンドを送ったり、アプリケーションプログラム本体APの実行結果データを連携インタフェース30に受け取ったりすることを行う。

【0098】また、チャネルポート401は、受信時のバッファリング機能を持ち、チャネル400や入出力チャネル403を通して送られてきたデータを読みだし要求があるまで保持する。

【0099】図7は、ルーティングマネージャ40の構

17

成を説明するための図である。

【0100】ルーティングマネージャ40は、処理装置16のメモリ15上におかれる管理テーブル作成プログラム41、接続要求プログラム42、接続管理プログラム43、移動管理プログラム44、接続待ちプログラム45の各プログラムと接続先管理テーブル46、サービスエンティティ管理テーブル47、接続管理テーブル48から構成される。

【0101】処理装置16のメモリ15上におかれたルーティングマネージャ40が起動されると、まず管理テーブル作成プログラム41が外部記憶装置18に格納されている接続先コンピュータの名称の一覧を読み込み、処理装置16のメモリ15上に接続先管理テーブル46を作成する。

【0102】次に、ルーティングマネージャ40の動作するコンピュータで動作するサービスエンティティの名称の一覧を読み込み、処理装置16のメモリ15上にサービスエンティティ管理テーブル47を作成する。

【0103】接続要求プログラム42は、管理テーブル作成プログラム41により処理装置16のメモリ15上に作成された接続先管理テーブル46に格納されたコンピュータ名の一覧の先頭から順に接続先コンピュータ名を取得し、その接続先コンピュータに対して接続を要求する。

【0104】接続ができない場合には、次の接続先コンピュータ名を取得し、接続ができるまで順次接続要求を行う。

【0105】また、接続先管理テーブルに格納された接続先コンピュータ名がなくなるまで、順次、接続要求を行っても接続できない場合、エラーメッセージを出力し、処理を終了する。

【0106】接続管理プログラム43は、接続要求プログラム42により接続されたコンピュータ（接続先コンピュータ）と接続が完了したことを受け取ると、接続先コンピュータに接続要求を行ったコンピュータ（接続元コンピュータ）のコンピュータ名を送信し、接続先コンピュータから接続先コンピュータ名が返信されてくるのを待つ。

【0107】接続先コンピュータから送られた接続先コンピュータ名を受け取ると、その接続先コンピュータの名称と接続先コンピュータに接続されたチャネル400のチャネルポート番号のペアを接続管理テーブル48に格納する。

【0108】接続待ちプログラム45がチャネルポート番号を受け取ったことにより制御が渡されてきた場合、そのチャネルポート番号から接続元コンピュータ名、もしくはサービスエンティティ10の名称が送られてくるのを待つ。

【0109】接続元コンピュータ名、もしくはサービスエンティティ10の名称を受け取るとその名称とその間

18

に接続された論理的通信路のチャネルポート番号のペアを接続管理テーブル48に格納し、送られてきた名称が接続元コンピュータ名の場合にはその接続元コンピュータにたいして接続先コンピュータ名を送信する。

【0110】移動管理プログラム44は、接続管理プログラム43が接続元コンピュータ名またはサービスエンティティ10の名称とチャネルポート番号のペアを接続管理テーブル48に格納する、ある一定時間の間に接続要求がなされなかった場合に起動される。

【0111】移動管理プログラム44は、接続管理テーブル48の先頭から順にチャネルポート番号を取得し、そのチャネルポート番号401に接続されたチャネル400からサービスエージェント20を読み出し、読み出したサービスエージェント20の移動先名204に指定されたサービスエンティティの名称が接続管理テーブル48に格納されているかどうかをチェックする。

【0112】格納されている場合には、そのサービスエージェント20をサービスエンティティ30に送る。

【0113】格納されていない場合には、接続されている他のルーティングマネージャ40を接続管理テーブル48から検索し、その結果得られたルーティングマネージャ40にそのサービスエージェント20を送る。

【0114】そして、チャネル400から読み出すサービスエージェント20がない場合には、次のチャネルポート番号401に接続されたチャネル400から同様にサービスエージェント20を読み出す。これを接続管理テーブル48に格納されたチャネルポート番号401がなくなるまで繰り返す。

【0115】接続管理テーブル48に格納されたチャネルポート番号の全てについて順次読み出しを行うと次の接続待ちプログラム45に制御を渡す。

【0116】接続待ちプログラム45は、ある一定時間他のルーティングマネージャ40もしくはサービスエンティティ10からの接続要求を待つ。

【0117】ある一定時間の間に接続要求があれば、その接続要求元のルーティングマネージャ40との間でチャネル400を接続するか、接続要求元のサービスエンティティ10との間で入出力チャネル403を接続し、新たな空きチャネルポート402を生成し、新たに接続されたチャネル400の接続されたチャネルポート401のチャネルポート番号を接続管理プログラム43に送る。

【0118】ある一定時間の間に接続要求がなければ、移動管理プログラム44に制御を渡す。

【0119】次に、その接続先管理テーブル46とサービスエンティティ管理テーブル47と接続管理テーブル48の構成について説明する。

【0120】図8(a)に接続先管理テーブル46の構成を、図8(b)にサービスエンティティ管理テーブル47の構成を、図8(c)に接続管理テーブル48の構

成の例を示す。

【0121】図8(a)に示す接続先管理テーブル46は、接続先のコンピュータの名称の一覧を保持する接続先コンピュータ名フィールド460を持つ。

【0122】接続先コンピュータ名フィールド460は、ルーティングマネージャ40の起動時に接続要求を行うコンピュータの名称であるコンピュータ名461を値(ここでは、C1とC3)として持つ。

【0123】図8(b)に示すサービスエンティティ管理テーブル47は、ルーティングマネージャ40の動作するコンピュータ上で動作するサービスエンティティ10の名称の一覧を保持するサービスエンティティ名フィールド470を持つ。

【0124】サービスエンティティ名フィールド470は、ルーティングマネージャ40の動作するコンピュータ上で動作するサービスエンティティ10の名称471を値(ここでは、SE3とSE4)として持つ。

【0125】図8(c)に示す接続管理テーブル48は、チャネル400により接続された他のルーティングマネージャ40の動作するコンピュータの名称や入出力チャネル403により接続されたサービスエンティティ10の名称を保持する接続先名称フィールド480と、それらの接続先との接続に用いられているチャネル400や入出力チャネル403が接続されているチャネルポート401のチャネルポート番号を保持するチャネルポート番号フィールド481と、接続先がルーティングマネージャ40であるかサービスエンティティ10であるかを区別するための接続先種別フィールド484とを持つ。

【0126】接続先フィールド480は、ルーティングマネージャ40にチャネル400や入出力チャネル403を用いて接続されている接続先名482を値(ここでは、C1とSE3とSE4)として持つ。

【0127】チャネルポート番号フィールド481はチャネル400や入出力チャネル403が接続されているチャネルポート番号483(ここでは、1と2と3)を接続先名482に対応付けて持つ。

【0128】接続先種別フィールド484は、チャネルの接続先がルーティングマネージャであることを示すルーティングマネージャ、サービスエンティティであることを示すサービスエンティティのどちらか一方の値を取る。この2つの値をあわせて接続先種別名485とする。ここでは、ルーティングマネージャとサービスエンティティである。

【0129】次に、本実施例の連携インタフェース30の構成について説明する。

【0130】図9は、本実施例の連携インタフェース30を実現するための処理装置16におけるプログラムの構造を示した図である。

【0131】本実施例の連携インタフェース30は、処

理装置16のメモリ15上におけるデータ選択プログラム31、データ出力プログラム32、実行制御プログラム33、データ取得プログラム34、データ格納プログラム35、移動先操作プログラム36、関数制御プログラム37の各プログラムと関数テーブル371、格納手続き管理テーブル38から構成される。

【0132】連携インタフェース30は、入出力チャネル403を通じてサービスエージェント20を受け取る、受け取ったサービスエージェント20をデータ作成プログラム31に渡す。

【0133】データ選択プログラム31は、受け取ったサービスエージェント20の中の入力データ選択手続き231を用いて、入力済みデータ項目27の中からアプリケーションプログラム本体APに渡すデータを選択しているデータ項目である入力データ項目271を選択する。

【0134】入力データ項目271が決定されると、サービスエージェント20の中の入力データの交換手続き232を用いて、入力データ項目271に格納されているデータからアプリケーションプログラム本体APに渡すデータ、もしくは画面に出力するデータを作成し、どちらのデータであるかのタグを付加し、データ出力プログラム32に作成したデータを渡す。

【0135】データ出力プログラム32にデータが渡されると、データ出力プログラム32は受け取ったデータのタグからデータを画面に出力するか、アプリケーションプログラム本体APに送るかを判定し、画面に出力する場合にはサービスエージェント(20)の入力データ表示手続き234を用いて画面に出力する。

【0136】アプリケーションプログラム本体APに渡す場合、データを実行制御プログラム33に渡す。

【0137】サービスエージェント20に記述された入力データ手続き234が、プログラムではなく単なる関数名だけの場合、データ出力プログラム32は、関数制御プログラム37にその関数名を渡す。

【0138】関数制御プログラム37は、関数名を受け取ると、関数テーブル371からその関数名に該当する関数を検索し、検索したプログラムを実行する。

【0139】実行制御プログラム33は、データ出力プログラム32からのデータを受け取る、サービスエージェント20の制御手続き26を実行してアプリケーションプログラム本体APの実行を制御する。

【0140】アプリケーションプログラム本体APを実行する際に、アプリケーションプログラム本体APにデータを渡す必要がある場合には、サービスエージェント20のパラメタ設定手続き233を用いてアプリケーションプログラム本体APにデータを送る。

【0141】アプリケーションプログラム本体APの実行を制御し、制御手続きの実行が終了すると、サービスエージェント20のデータ取得手続き233を用いてア

21

アプリケーションプログラム本体APの実行結果データを取得し、取得したデータをデータ取得プログラム34に渡す。

【0142】制御手続き26が関数名のみであったり、制御手続き26の中に関数名のみの処理が含まれている場合には、その関数名を関数制御プログラム37に渡し、該当する関数のプログラムを実行する。

【0143】データ取得プログラム34は、実行結果データを受け取るとサービスエージェンツ20の入力データ選択手続き221を用いてサービスエージェンツ20の全データ項目21の中からアプリケーションプログラム本体APの実行結果データを格納するデータ項目である出力データ項目281を選択する。

【0144】画面が出力されている場合には、サービスエージェンツ20の出力データ取得手続き222を用いて画面に入力されたデータのうち、必要なデータのみを取得する。

【0145】これらの取得したデータ（アプリケーションプログラム本体APの実行結果データ、画面からの入力データ）と出力データ項目281を、データ格納プログラム35に送す。

【0146】サービスエージェンツ20に記述された出力データ取得手続き222が、プログラムではなく単なる関数名だけの場合、データ取得プログラム34は、関数制御プログラム37にその関数名を渡し、該当する関数名の関数のプログラムを実行する。

【0147】データ格納プログラム35は、データ取得プログラム34からのデータと出力データ項目281を受け取る。サービスエージェンツ20の出力データ処理手続き24を参照し、出力データ処理手続き24があれば、その出力データ処理手続き24を用いて、受け取ったデータ及び全項目データ21に格納されているデータを処理し、処理後のデータを出力データ変換手続き224を用いて、受け取った出力データ項目281のデータ形式に変換し、変換したデータを出力データ項目281のそれぞれのデータ項目に格納する。

【0148】出力データ処理手続き24がない場合には、出力データ変換手続き224を用いて、受け取った出力データ項目281のデータ形式に変換し、変換したデータを出力データ項目281のそれぞれのデータ項目に格納する。

【0149】移動先操作プログラム36は、出力データ項目281にデータが格納されると呼び出され、移動先リスト202の一致から先頭要素を取りだしそれを移動先名204に格納し、移動先名204を更新したサービスエージェンツ20を入力データチャネル403を通してルーティングマネージャ40に送る。

【0150】また、特殊な場合として、サービスエージェンツ20の構成要素をサービスエージェンツを受け取ったサービスエージェンティ10もしくはサービスエ

22

ージェンティ10の動作するコンピュータCに応じて変更する場合は、その変更する構成要素を、図10に示すような条件判定を含む記述形式で記述し、格納しておく。

【0151】上記各プログラム（31～35）では、上述したように、サービスエージェンツ20の中の該当する手続きを用いて処理を行うが、その際、そのサービスエージェンツ20の中の該当する手続きが、図10に示すような、条件判定を含む形式で記述されている場合には、手続きを実行する前に、まず、取得した条件判定を含む手続き1801を実行する。

【0152】この条件判定を含む手続きは、連携インタフェース30が持っている情報である、動作しているコンピュータ名及びサービスエージェンティ名（これらの各値はサービスエージェンティ起動時に取得される）の各々の値を用いて条件1802を評価する。

【0153】評価結果が真であれば、手続き1803を実行する。偽であれば、手続き名1804に相当する手続きをサービスエージェンティ10もしくは関数テーブル371から該当する手続き名を持つ手続きを検索し、その手続きを実行する。

【0154】サービスエージェンティ10やサービスエージェンティ10の動作するコンピュータに応じて処理を変更しない場合には、処理手続き1803のみが記述される。

【0155】これにより、サービスエージェンツ20の構成要素をサービスエージェンティ10もしくは、コンピュータに応じて変更することが可能になる。

【0156】さらに、データ出力プログラム32において、図11に示すようなグラフィカルユーザインタフェース（ウィンドウ、ボタンウィンドウ、ラベルウィンドウ、テキスト入力ウィンドウ、テキスト出力ウィンドウ、ビットマップウィンドウ、画像表示ウィンドウ、画面表示ウィンドウ、ライン）を動的に構成し、データを表示するようにしてもよい。

【0157】この場合は、入力データ表示手続き234に、入力データ項目にそのデータ項目の値をグラフィカルユーザインタフェースに表示するための組み合わせ手続きを付与したものを格納する。

【0158】図12（a）は、その入力データ項目と組み合わせ手続きの記述形式を示した図である。

【0159】図12（a）に示すように、入力データ項目とそのデータ項目の値をグラフィカルユーザインタフェースに表示するための組み合わせ手続きは、入力データ項目の項目名1601とその項目名に格納されている値を表示するための組み合わせ手続き1602のペアとして、入力データ表示手続き234中に記述される。

【0160】項目名に関係なく画面に出力する場合には、項目名にdefaultを指定する。

【0161】この場合、データ出力プログラム32は、入力データ表示手続き234を受け取る項目名160

50

23

1と組み合せ手続き1602のペアを先頭から取り出し、その項目名1601がdefaultであれば、その取り出した組み合せ手続き1602を実行し、グラフィカルユーザインタフェースを作成する。

【0162】項目名1601がdefault以外であれば、項目名1601に指定された入力データ項目の値を組み合せ手続きを用いて実行し、組み合せ手続きに指定されたようにグラフィカルユーザインタフェースを構成し、データを出力する。

【0163】図12(b)は、図11に示すグラフィカルユーザインタフェースの入力データ表示手続き234の1例を示した図である。

【0164】この時、入力データ項目は項目1と項目2であり、それぞれの項目の値は、リスト(a b c)と数値12であるとする。

【0165】図12(a)に示した例では、データ出力プログラム32は、図12(b)に示したような入力データ表示手続きを受け取ると、まず、その先頭の項目名と組み合せ手続きのペア1621を取得する。

【0166】項目名と組み合せ手続きのペア1621の項目名がdefaultであるので、その組み合せ手続きを実行する。

【0167】この時の組み合せ手続きは、ウィンドウ表示関数1611であるので、この関数を実行することにより、図11のウィンドウ1501が表示装置17に表示される。表示したウィンドウ1501の識別子を変数idに格納される。

【0168】次に、データ出力プログラム32は、次の項目名と組み合せ手続きのペア1622を取得する。

【0169】このペア1622の項目が項目1であるので、項目1の値を用いて、組み合せ手続きを実行する。

【0170】この組み合せ手続きは、まず項目1の値が空が否かチェックされ、空でなければその先頭要素aを取り出し、それを変数idの値aをボタンウィンドウ表示関数1612を用いてウィンドウ1501上にボタン1502を表示する。表示されたボタンの識別子は変数idに格納される。

【0171】次に、y座標の表示位置を更新し、ボタン1502が押されたときの処理を関数set-scriptを用いて登録し、項目1の先頭要素を除いた要素(b c)を項目1の値とする。

【0172】同様にして、項目1の値が空になるまでボタンウィンドウ表示関数1612を用いてウィンドウ1501上にボタン1502、1504を表示する。

【0173】次に、データ出力プログラム32は、次の項目名と組み合せ手続きのペア1623を取得する。このペア1623の項目名が項目2であるので、項目2の値を用いて、組み合せ手続きを実行する。

【0174】この組み合せ手続きは、テキスト入力ウィンドウ表示関数であるので、この関数を用いて項目2の

24

値をウィンドウ1501上のテキスト入力ウィンドウ1506に表示する。表示されたテキスト入力ウィンドウ1506の識別子は変数id2に格納される。

【0175】次に、データ出力プログラム32は、次の項目名と組み合せ手続きのペア1624を取得する。

【0176】このペア1624の項目名がdefaultであるので、組み合せ手続きを実行し、ウィンドウ1501上にボタン1505を表示し、その識別子を変数id3に格納し、このボタンが押されたときの処理手続き1633を関数set-scriptは指定されたidと処理手続きを図13に示す操作処理手続き管理テーブル39の識別子フィールド1701、操作処理手続きフィールド1702にそれぞれ格納する。

【0177】この操作処理手続き管理テーブル39は、識別子を格納する識別子フィールド1701とその識別子のウィンドウ基本部品が選択されたときに実行されるプログラムを格納する操作処理手続きフィールド1702からなる。

【0178】入力データ表示手続き234の実行が終了すると、実行制御プログラム33に制御を渡す。

【0179】実行制御プログラム33では、サービスエージェント20の制御手続き26を実行して画面の入出力の実行制御をする。表示装置17に表示されたグラフィカルユーザインタフェースの基本部品が、マウス12により選択されると、ウィンドウシステムを経由してマウス12により選択された基本部品の識別子が実行制御プログラム33に伝えられる。

【0180】実行制御プログラム33は、ウィンドウシステムからの識別子を受け取ると、受け取った識別子が操作処理手続き管理テーブル39の識別子フィールド1701に格納された識別子1703の中に存在するかを検査し、存在すればそのプログラム1704を実行する。

【0181】受け取った識別子が操作処理手続き管理テーブル39の識別子フィールド1701に存在しなければ、なにもしないで次の識別子がウィンドウシステムから送られるのを待つ。

【0182】ウィンドウシステムから表示装置17に表示した画面の終了要求が発生すると、データ取得プログラム34に制御を渡す。

【0183】データ取得プログラム34は、サービスエージェント20の出力データ取得手続き222を用いて画面に入力されたデータの中で必要なデータを取得し、操作処理手続き管理テーブル39をクリアし、取得したデータと出力データ項目281をデータ格納プログラムに渡す。

【0184】このグラフィカルユーザインタフェース表示手続きを設けることにより、利用者の画面の入力における操作性を向上できる。

【0185】次に、その関数テーブル371と移動先リ

25

スト202と移動リスト203について説明する。

【0186】図14(a)に関数制御プログラム377の使用する関数テーブル371の構造を示す。

【0187】図14(a)に示すように、関数テーブル371は、関数名フィールド372と関数の処理プログラム本体である処理フィールド373とからなる。

【0188】処理フィールド373は、関数の処理内容を記述したプログラム373a、または、関数の処理内容を記述したプログラムの格納位置373bのいずれかが記述される。

【0189】関数制御プログラム377は、該当する関数名の処理フィールド373がプログラム373aであれば、そのプログラムを実行する。

【0190】一方、プログラムの格納位置373bである場合は、その格納位置にあるプログラム374をネットワーク0を通じて処理装置16のメモリ15上に読み込んだ後、そのプログラムを実行する。

【0191】プログラムの格納位置373bは、図14(b)に示す記述形式により指定される。

【0192】プログラムの格納位置373bは、プログラムの格納されているコンピュータの名称375aと、コンピュータ名375aで指定されたコンピュータ上のプログラムの格納位置を表す格納パス375cと、コンピュータ名375aと格納パス375bを区別するための区切り記号375bとを用いて、図14(b)に示すように、コンピュータ名375a、区切り記号:375b、格納パス375cの形式で指定する。

【0193】図15(a)は、サービスエージェント20の移動先リスト202の記述形式を、図15(b)は、その具体的記述例を示した図である。

【0194】移動先リスト202は、サービスエージェント20を移動する先のコンピュータの名称である移動先コンピュータ名912、移動先のコンピュータで利用するサービスエンティティ10の名称であるサービスエンティティ名913の組みである移動先911を、サービスエージェント20の移動順序に従って矢印(→)でつないだものである。

【0195】移動先のコンピュータ名が分からない場合には、移動先コンピュータ名912にUnknownを指定する。

【0196】図15(b)に示すように、サービスエージェント20は、C1という名称のコンピュータのSE1という名称のサービスエンティティ10の持つアプリケーションプログラム本体AP1を用いて処理を行った後、C2という名称のコンピュータのSE2という名称のサービスエンティティ10の持つアプリケーションプログラム本体AP2を用いて処理を行い、次にC3という名称のコンピュータのSE3という名称のサービスエンティティ10の持つアプリケーションプログラム本体AP3を用いて処理を行い、最後にサービスエンティティ

35

10(SE4)の動作するコンピュータ名は不明であるが、SE4という名称のサービスエンティティ10の持つアプリケーションプログラム本体AP4を用いて処理を行うことを示している。

【0197】図16(a)は、サービスエージェント20の移動リスト203の記述形式を、図16(b)は、その具体的記述例を示した図である。

【0198】図16(a)に示すように、移動リスト203は、移動先コンピュータ名912がUnknownの時に用いられる。

【0199】移動先コンピュータ名912がUnknownの場合、各コンピュータのルーティングマネージャ40は、サービスエージェント20の移動先名204に指定されている名称を持つサービスエンティティ10が存在するか否かをチェックし、存在しない場合、そのコンピュータの名称を移動コンピュータ名102に格納し、他のコンピュータ上のルーティングマネージャ40にサービスエージェント20を送る。

【0200】移動リスト203は、移動先コンピュータ名912がUnknownの場合に、サービスエージェント20の移動先名204に指定されている名称をサービスエンティティ10が存在しないコンピュータの名称をサービスエージェント20が移動してきた順序に従って矢印(→)で繋いだものである。

【0201】移動リスト203は、サービスエージェント20が、移動するたびにそのコンピュータ名が順次、追加されていき、サービスエージェント20の移動先名204に指定されているサービスエンティティ10が動作するコンピュータが見つかった場合にクリアされる。

【0202】図16(b)の例では、サービスエージェント20の移動先名204に指定されているサービスエンティティ10が、C1という名称のコンピュータに存在せず、次のC2という名称のコンピュータにも存在しなかったことを示している。

【0203】次に、本実施例のサービスエージェント20を用いたアプリケーションプログラム間の連携処理を実現するためにルーティングマネージャ40が行うサービスエージェント20の移動処理のフローを図17～図20に示す。

【0204】サービスエージェント20は、それが作成されたコンピュータのルーティングマネージャ40に送られる。この時点で、サービスエージェント20の移動先リスト202の値は図14(b)に示した移動順序であるとする。この時のサービスエージェント20の移動先名204の値は、移動先リスト202の先頭の要素SE1(911a)となっている。

【0205】本実施例のサービスエージェント20の移動処理は、ルーティングマネージャ40は、サービスエージェント20を受け取る(ステップ1100)、まず、サービスエージェント20の移動先名204の移動

27

先コンピュータ名がルーティングマネージャ40の動作しているコンピュータ名と同じ否かをチェックする(ステップ1101)。

【0206】サービスエージェント20の移動先名204の移動先コンピュータ名とルーティングマネージャ40の動作しているコンピュータ名が異なる場合、接続管理テーブル48の接続先種別フィールド484の接続先種別名485がルーティングマネージャである接続先の中から、接続先名称フィールド480の接続先名482が移動先コンピュータ名と同じ接続先が登録されている

10

可否かをチェックする(ステップ1102)。

【0207】移動先コンピュータ名が登録されている場合、その移動先コンピュータ名に該当する接続先名482のチャネルポート番号フィールド481に格納されているチャネルポート番号483を取りだし(ステップ1103)、そのチャネルポート番号483に接続されているチャネル400を通して移動先コンピュータのルーティングマネージャ40にサービスエージェント20を送る(ステップ1104)。

20

【0208】登録されていない場合、接続管理テーブル48の接続先種別フィールド484の接続先種別名485がルーティングマネージャである接続先の中から、最初に登録されている接続先名482のチャネルポート番号フィールド481に格納されているチャネルポート番号483を取りだし(ステップ1105)、そのチャネルポート番号に接続されているチャネル400を通してチャネルに接続されている登録先コンピュータのルーティングマネージャ40にサービスエージェント20を送る(ステップ1106)。

【0209】サービスエージェント20の移動先名204の移動先コンピュータ名912とルーティングマネージャ40の動作しているコンピュータ名と同じである場合、接続管理テーブル48の接続先種別フィールド484に格納されている接続先種別名485がサービスエンティティである接続先の中から接続先名称フィールド480の接続先名482が移動先名204のサービスエンティティ名913と同じサービスエンティティが登録されている可否かをチェックする(ステップ1107)。

【0210】登録されていれば、その登録されているサービスエンティティのチャネルポート番号フィールド481に格納されているチャネルポート番号483を取得し(ステップ1108)、そのチャネルポート番号483に接続されている入出力チャネル403を通して、サービスエンティティ20の連携インタフェース30にサービスエージェント20を送る(ステップ1109)。

【0211】サービスエンティティ名が接続管理テーブル48に登録されていないければ、サービス管理テーブル47にサービスエンティティ名が登録されている可否かをチェックする(ステップ1110)。

【0212】サービス管理テーブル47にサービスエン

28

ティティ名が登録されていれば、そのサービスエンティティ10を起動する(ステップ1111)。

【0213】起動されたサービスエンティティ10の連携インタフェース30は、ルーティングマネージャ40に接続要求を行う。ルーティングマネージャ40は、サービスエンティティ10の連携インタフェース30からの接続要求を待ち(ステップ1112)、接続要求を受け取るとルーティングマネージャ40と連携インタフェース30間の入出力チャネル403を生成し(ステップ1113)、起動したサービスエンティティ名とその入出力チャネル403のチャネルポート番号を接続管理テーブル48に登録し(ステップ1114)、生成した入出力チャネル403を通してサービスエージェント20をサービスエンティティ10の連携インタフェース30に送る(ステップ1109)。

【0214】サービス管理テーブル47にサービスエンティティ名が登録されていなければ、サービスエージェント20の移動先名204の移動先コンピュータ名912をUnknownに変更し(ステップ1115)、接続管理テーブル48の接続先種別フィールド484の接続先種別名485がルーティングマネージャである接続先の中から、最初に登録されている接続先名482のチャネルポート番号フィールド481に格納されているチャネルポート番号483を取りだし(ステップ1105)、そのチャネルポート番号に接続されたチャネル400を通してチャネルに接続されている登録先コンピュータのルーティングマネージャ40にサービスエージェント20を送る(ステップ1106)。

【0215】サービスエージェント20の移動先名204の移動先コンピュータ名912がUnknownの場合、移動先名204のサービスエンティティ名が、接続管理テーブル48の接続先種別フィールド484の接続先種別名485がサービスエンティティである接続先の中から接続先名称フィールド480の接続先名482がサービスエージェント20の移動先名204のサービスエンティティ名と同じ接続先が登録されている可否かをチェックする(ステップ1115)。

【0216】登録されていれば、その登録されているサービスエンティティのチャネルポート番号フィールド481に格納されているチャネルポート番号483を取得し(ステップ1116)、取得したチャネルポート番号483に接続されている入出力チャネル403を通して、サービスエンティティ20の連携インタフェース30にサービスエージェント20を送る(ステップ1117)。

【0217】サービスエンティティ名が接続管理テーブル48に登録されていないければ、次にサービス管理テーブル47にサービスエンティティ名が登録されている可否かをチェックする(ステップ1118)。

【0218】サービス管理テーブル47にサービスエン

29

ティティ名が登録されていれば、そのサービスエンティティ10を起動する(ステップ1119)。

【0219】起動されたサービスエンティティ10の連携インタフェース30は、ルーティングマネージャ40に接続要求を行う。ルーティングマネージャ40は、サービスエンティティ10の連携マネージャ30からの接続要求を待ち(ステップ1120)、接続要求を受け取る。ルーティングマネージャ40と連携インタフェース30間の入出力チャネル403を生成し(ステップ1121)、起動したサービスエンティティ名とその入出力チャネル403のチャネルポート番号を接続管理テーブル48に登録し(ステップ1122)、生成した入出力チャネル403を通してサービスエージェント20をサービスエンティティ10の連携インタフェース30に送る(ステップ1117)。

【0220】接続管理テーブル48、サービス管理テーブル47のいずれのテーブルにも登録されていなければ、接続管理テーブル48の接続先種別フィールド484の接続先種別名485がルーティングマネージャである接続先があるかをチェックする(ステップ1123)。

【0221】接続先が登録されていれば、その中の最初に登録されている接続先名482を取りだし(ステップ1124)、その接続先名482が移動リスト203に移動コンピュータ名102にあるかをチェックする(ステップ1125)。

【0222】移動リスト203になければ、移動リスト203の移動コンピュータ名102に現在のコンピュータの名称を追加し(ステップ1126)、追加した移動リスト203をサービスエージェント20に格納し(ステップ1127)、その接続先名482のチャネルポート番号フィールド481に格納されているチャネルポート番号483を取りだし(ステップ1128)、そのチャネルポート番号に接続されたチャネル400を通してチャネルに接続されているコンピュータのルーティングマネージャ40にサービスエージェント20を送る(ステップ1129)。

【0223】移動リスト203にあれば、接続管理テーブル48の接続先種別フィールド484の接続先種別名485がルーティングマネージャである次の接続先があるかをチェックする(ステップ1123)。

【0224】次の接続先が接続管理テーブル48に登録されていれば、その接続先名482を取りだし(ステップ1124)、その接続先名482がサービスエージェント20に格納されている移動リスト203の移動コンピュータ名102にあるかをチェックする(ステップ1125)。

【0225】これを接続管理テーブル48の接続先種別フィールド484の接続先種別名485がルーティングマネージャである接続先がなくなるか、移動リスト20

30

3にない接続先名482が見つかるまで順次繰り返す。

【0226】その結果、接続管理テーブル48の接続先種別フィールド484の接続先種別名485がルーティングマネージャである接続先がなくなった場合、エラーとしてそのサービスエージェント20の移動を終了する(ステップ1130)。

【0227】サービスエージェント20を受け取ったサービスエンティティ10の連携インタフェース30は、受け取ったサービスエージェント20に記述されているデータ入力手続き23を実行しサービスエージェント20からサービスエンティティ10に送るデータを取得し(ステップ1131)、制御手続き26を実行して、そのサービスエンティティ10の持つアプリケーションプログラム本体A Pの実行を制御し(ステップ1132)、その実行結果データをサービスエージェント20のデータ出力手続き22を実行してサービスエージェント20に格納し(ステップ1133)、サービスエージェント20の移動手続き25を実行して移動先名204の値を次の移動先に書き換え(ステップ1134)、移動先名204の値を書き換えたサービスエージェント20を、連携インタフェース30からルーティングマネージャ40へ入出力チャネル403を通して送る(ステップ1135)。

【0228】ルーティングマネージャ40は、連携インタフェース30から送られたサービスエージェント20を入出力チャネル403を通して受け取る(ステップ1100)、チャネル400を通してサービスエンティティ20を受け取ったときと同様に、移動先名204の移動先コンピュータ名912をチェックし(ステップ1101)、サービスエージェント20を次の移動先にチャネル400を通して送る(ステップ1104)。

【0229】このサービスエージェントの移動処理をサービスエージェント20の移動先リスト202の最後の要素になるまで、ルーティングマネージャ10間で繰り返すことにより、サービスエージェント20をサービスエンティティ10間で移動させ、移動させたサービスエージェント20をサービスエンティティ10の連携インタフェース30で処理することにより各アプリケーションプログラム本体A Pの実行を制御する。

【0230】次に、図21～図24にサービスエンティティ10の連携インタフェース30において、受け取ったサービスエージェント20の持つデータや手続きを利用しながら、アプリケーションプログラム本体A Pの実行を制御するための処理フローを示す。

【0231】連携インタフェース30は、入出力チャネル403を通じてルーティングマネージャ10からサービスエージェント20を受け取る(ステップ1200)、受け取ったサービスエージェント20の入力データ選択手続き231を実行して入力済みデータ項目27の中からアプリケーションプログラム本体A Pに渡すデ

ータを格納しているデータ項目である入力データ項目271を決定する(ステップ1201)。

【0232】入力データ項目271が決定されると、サービスエージェント20の入力データ変換手続き232がプログラムであるか関数名であるかをチェックする(ステップ1202)。

【0233】入力データ変換手続き232がプログラムであれば、その入力データ変換手続き232がアプリケーションプログラム本体A Pへのデータ入力関数か画面出力関数かをチェックし(ステップ1203)、データ入力関数であれば、その関数を実行して、入力データ項目271に格納されているデータからアプリケーションプログラム本体A Pに渡すデータを生成する(ステップ1204)。

【0234】画面出力関数であれば、その関数を実行して、入力データ項目271に格納されているデータから画面に出力するデータを作成する(ステップ1205)。

【0235】そして、生成されたデータには、どちらのデータであるかを示すタグが付加される(ステップ1206)。

【0236】入力データ変換手続き232が関数名であれば、その関数名のプログラムを関数制御プログラム37を通して関数テーブルから取得し(ステップ1207)、取得した関数がアプリケーションプログラム本体A Pへのデータ入力関数か画面出力関数かをチェックする(ステップ1203)。

【0237】タグが付加されたデータは、タグからデータを画面に出力するか、アプリケーションプログラム本体A Pに渡すかが判定され(ステップ1208)、画面に出力するデータの場合にはサービスエージェント20の入力データ手続き234がプログラムであるか関数名であるかがチェックされる(ステップ1209)。

【0238】入力データ手続き234がプログラムであれば、その入力データ手続き234を実行し、画面出力データを画面に出力する(ステップ1210)。

【0239】入力データ手続き234が関数名であれば、その関数名のプログラムを関数制御プログラム37を通して関数テーブルから取得し(ステップ1211)、取得した関数を実行し、画面出力データを画面に出力する(ステップ1210)。

【0240】アプリケーションプログラム本体A Pに渡すデータの場合、そのデータのタグを削除する(ステップ1212)。

【0241】次に、サービスエージェント20の制御手続き26がプログラムであるか関数名であるかがチェックされる(ステップ1213)。

【0242】制御手続き26がプログラムであれば、その制御手続き26を実行することにより、アプリケーションプログラム本体A Pの実行を制御する(ステップ1

214)。

【0243】制御手続き26が関数名であれば、その関数名のプログラムを関数制御プログラム37を通して関数テーブルから取得し(ステップ1215)、取得した関数を実行することにより、アプリケーションプログラム本体A Pの実行を制御する(ステップ1214)。

【0244】制御手続き26によりアプリケーションプログラム本体A Pの実行制御が終了すると、サービスエージェント20のデータ取得手続き223がプログラムであるか関数名であるかがチェックされる(ステップ1216)。

【0245】データ取得手続き223がプログラムであれば、そのデータ取得手続き223を実行し、アプリケーションプログラム本体A Pの実行結果データを取得する(ステップ1217)。

【0246】データ取得手続き223が関数名であれば、その関数名のプログラムを関数制御プログラム37を通して関数テーブルから取得し(ステップ1218)、取得した関数を実行し、アプリケーションプログラム本体A Pの実行結果データを取得する(ステップ1217)。

【0247】画面が出力されている場合には、サービスエージェント20の制御手続き26がプログラムであるか関数名であるかがチェックされる(ステップ1219)。

【0248】制御手続き26がプログラムであれば、その制御手続き26を実行することにより、画面からの入出力を制御する(ステップ1220)。

【0249】制御手続き26が関数名であれば、その関数名のプログラムを関数制御プログラム37を通して関数テーブルから取得し(ステップ1221)、取得した関数を実行することにより、画面からの入出力を制御する(ステップ1220)。

【0250】画面が出力されている場合には、データ取得手続き223がプログラムであるか関数名であるかがチェックされる(ステップ1222)。

【0251】データ取得手続き223がプログラムであれば、そのデータ取得手続き223を実行し、画面からの入力データを取得する(ステップ1223)。

【0252】データ取得手続き223が関数名であれば、その関数名のプログラムを関数制御プログラム37を通して関数テーブルから取得し(ステップ1224)、取得した関数を実行し画面からの入力データを取得する(ステップ1223)。

【0253】その後、サービスエージェント20の出力データ選択手続き221がプログラムであるか関数名であるかがチェックされる(ステップ1225)。

【0254】出力データ選択手続き221がプログラムであれば、その出力データ選択手続き221を実行し、サービスエージェント20の全データ項目21の中から

33

アプリケーションプログラム本体A Pの処理結果データを格納するデータ項目である出力データ項目281を選択する(ステップ1226)。

【0255】出力データ選択手続き221が関数名であれば、その関数名のプログラムを関数制御プログラム37を通して関数テーブルから取得し(ステップ1227)、取得した関数を実行し、サービスエージメント20の全データ項目21の中からアプリケーションプログラム本体A Pの処理結果データを格納するデータ項目である出力データ項目281を選択する(ステップ1226)。

【0256】そして、サービスエージメント20の出力データ処理手続き24がプログラムであるか関数名であるかがチェックされる(ステップ1228)。

【0257】出力データ処理手続き24がプログラムであれば、その出力データ処理手続き24を実行し、受け取ったデータ及び全項目データ21に格納されているデータを処理する(ステップ1229)。

【0258】出力データ処理手続き24が関数名であれば、その関数名のプログラムを関数制御プログラム37を通して関数テーブルから取得し(ステップ1230)、取得した関数を実行し、受け取ったデータ及び全項目データ21に格納されているデータを処理する(ステップ1229)。

【0259】次に、サービスエージメント20の出力データ変換手続き224がプログラムであるか関数名であるかがチェックされる(ステップ1231)。

【0260】出力データ変換手続き224がプログラムであれば、その入力データ変換手続き224を実行し、受け取ったデータをそれに該当する出力データ項目281のデータ形式に変換し(ステップ1232)、変換したデータを出力データ項目281のそれぞれのデータ項目に格納する(ステップ1234)。

【0261】出力データ変換手続き224が関数名であれば、その関数名のプログラムを関数制御プログラム37を通して関数テーブルから取得し(ステップ1233)、取得した関数を実行し受け取ったデータをそれに該当する出力データ項目281のデータ形式に変換し(ステップ1232)、変換したデータを出力データ項目281のそれぞれのデータ項目に格納する(ステップ1234)。

【0262】出力データ項目281にデータが格納されると、サービスエージメント20の移動先リスト202の先頭要素を取りだし(ステップ1232)、それと現ポジションである移動先名204に格納し(ステップ1233)、移動先リスト202の値を更新し(ステップ1234)、更新した状態20を入出力チャネル403を通してルーティングマネージャ40に送る(ステップ1235)。

【0263】出力データ項目281にデータが格納され

34

ると、サービスエージメント20の移動手続き25がプログラムであるか関数名であるかがチェックされる(ステップ1235)。

【0264】移動手続き25がプログラムであれば、その移動手続き25を実行し、移動先名204の値を次の移動先に書き換え(ステップ1236)、移動先名204の値を書き換えたサービスエージメント20を、連携インタフェース30からルーティングマネージャ40へ入出力チャネル403を通して送る(ステップ1237)。

【0265】移動手続き25が関数名であれば、その関数名のプログラムを関数制御プログラム37を通して関数テーブルから取得し(ステップ1238)、取得した関数を実行し、移動先名204の値を次の移動先に書き換え(ステップ1236)、移動先名204の値を書き換えたサービスエージメント20を、連携インタフェース30からルーティングマネージャ40へ入出力チャネル403を通して送る(ステップ1237)。

【0266】したがって、上述したように、アプリケーションプログラム間の連携処理手続きをサービスエージメントというアプリケーションプログラム本体とは独立した形式で与え、アプリケーションプログラム本体に連携処理のための連携インタフェースを付加し、この連携インタフェースがサービスエージメントの持つ利用手続きを実行することによりアプリケーションプログラム間の連携処理をサービスエージメントを通して容易に変更することができる。

【0267】また、業務に応じて、その業務を進めるために必要なさまざまなアプリケーションプログラムとその操作を組み合わせた利用手順を、サービスエージメントの中に一連の作業手順と各作業で使用するアプリケーションプログラムの利用手続きとして記述し、このサービスエージメントをコンピュータネットワークに接続されたコンピュータ上で動作している各種サービスエントティの間で移動させ、サービスエージメントを受信したサービスエントティはサービスエージメントに記述された利用手続きを実行することにより、あらかじめアプリケーションプログラム間の連携処理手続きを記述しておかなくても、個々の利用者が自分自身の利用方法に適するアプリケーションプログラム間の連携処理を行えることが可能となる。

【0268】次に、上述した本実施例の分散システムの適用例としてコンピュータ統合生産システムを取り上げ、以下に詳細に説明する。

【0269】一般的に、コンピュータ統合生産システムは、製造・検査機器、それら機器の制御用アプリケーションプログラム、生産管理に必要なアプリケーションプログラムを製造物ごとに決められた順序で連携させる必要がある。

【0270】図25は、本適用例のコンピュータ統合生

産システムの構成を示す図である。

【0271】本適用例では、コンピュータ統合生産システムが各製造工程A、Bを管理するコンピュータC20、C30、各製造工程A、Bに接続された製造装置MA1、MB1、とこれらの製造装置MA1、MB1を制御するアプリケーションプログラムAP20、AP30、コンピュータC10上で製造指示書を作成するアプリケーションプログラムAP10、コンピュータを接続するネットワーク0から構成される。

【0272】また、本適用例では、コンピュータC10上のアプリケーションプログラムAP10で製造指示書を作成し、各製造工程に接続された製造機器をそれらの制御アプリケーションプログラムを通して連携制御することにより、製造を進めていく場合を取り挙げる。

【0273】本適用例によるコンピュータ統合生産システムでは、これらの各名アプリケーションプログラムAP10、AP20、AP30を有するサービスエンティティをそれぞれSE10、SE20、SE30とし、製造物に添付される製造指示書は、サービスエージェントSA2000とする。

【0274】そのサービスエージェントSA2000は、サービスエンティティSE10を用いて利用者を作成され、まず製造工程Aの製造機器MA1を制御する制御アプリケーションプログラムAP20を持つサービスエンティティSE20を通してアプリケーションプログラムAP20に製造指示を伝え、次に製造工程Bの製造機器MB1を制御する制御アプリケーションプログラムAP30を持つサービスエンティティSE30を通してアプリケーションプログラムAP30に製造指示を伝え、その後サービスエージェントSA2000を作成したサービスエンティティSE10に戻るものとする。

【0275】図26に、利用者がサービスエンティティSE10を用いてコンピュータC10上で作成した時点の製造指示書であるサービスエージェントSA2000の構造を示す。

【0276】図26に示すサービスエージェントSA2000は、識別子2001(ID1)、移動先リスト2002((C20 SE20)(C30 SE30)(C10 SE10))、移動先リスト2003(n11)、移動先名2004(C20SE20)に加え、全データ項目として、製造機器MA1の設定パラメータであるA工程製造速度項目2005(その値10)、製造機器MA1から取得するデータの項目であるA工程所要時間項目2006、製造機器MB1の設定パラメータである項目であるB工程製造速度項目2007(その値15)、製造機器MB1から取得するデータの項目であるB工程所要時間項目2008を持つ。

【0277】さらに、データ出力手続きとして、データ取得手続き2011、出力データ選択手続き2012、出力データ格納手続き2013を有し、データ入力手続

きとして、入力データ選択手続き2021、入力データ変換手続き2022、パラメータ設定手続き2023を、さらに制御手続き2030を有する。

【0278】このように、サービスエージェントSE10を用いて生成された上述したサービスエージェントSA2000は、チャネル400を通してコンピュータC10のルーティングマネージャRM10に送られる。

【0279】ルーティングマネージャRM10は、サービスエージェントSA2000の移動先名2004の値(C20 SE20)をチェックし、移動先名に指定されたコンピュータC20との間に接続されたチャネル400を検索し、その結果得られたチャネル400を通してコンピュータC20にサービスエージェントSA2000を送る。

【0280】コンピュータC20上のルーティングマネージャRM20は、送られてきたサービスエージェントSA2000を受け取る、その移動先名2004の値(C20 SE20)をチェックし、移動先名に指定されたサービスエンティティSE20との間に接続されたチャネル400を検索し、その結果得られたチャネル400を通してサービスエンティティSE20にサービスエージェントSA2000を送る。

【0281】サービスエンティティSE20は、チャネル400を通してサービスエージェントSA2000を受け取ると、サービスエンティティSE20の連携インタフェース30がサービスエージェントSA2000の持つ入力データ選択手続き2021を取得し、その手続き2021を実行し、サービスエンティティSE20の持つアプリケーションプログラムAP20に渡すべきデータ項目であるA工程製造速度項目2005を決定する。

【0282】その後、サービスエージェントSA2000の持つ入力データ変換手続き2022を取得し、その手続き2022を実行し、選択された項目であるA工程製造速度項目2005の値10を取り出し、そのデータがアプリケーションプログラムに渡すデータであることを示すタグを付加した後、サービスエージェントSA2000の持つパラメータ設定手続き2023を取得し、その手続き2023を実行し、アプリケーションプログラムAP20に指定された値10を設定する。

【0283】そして、サービスエージェントSA2000の持つ制御手続き2030を取得し、その手続き2030を実行することによりアプリケーションプログラムAP20の処理を行なう。

【0284】アプリケーションプログラムAP20の処理が終了すると、サービスエージェントSA2000の持つデータ取得手続き2011を取得し、その手続き2011を実行し、アプリケーションプログラムの実行結果を取得する。

【0285】その後、サービスエージェントSA2000

37

0の持つ出力データ選択手続き2012を取得し、その手続き2012を実行し、アプリケーションプログラムの実行結果を格納するデータ項目であるA工程所要時間2006を選択し、サービスエージェントSA2000の持つ出力データ格納手続き2013を取得し、その手続き2013を実行し、アプリケーションプログラムAP200の実行結果をA工程所要時間項目2006に格納する。

【0286】次に、移動先リスト2002の2番目の要素である(C30 SE30)を取り出し、それを移動先名2004に格納し、チャネル400を通してコンピュータC20上のルーティングマネージャRM20に送る。

【0287】ルーティングマネージャRM20は、サービスエンティティSE20からチャネル400を通して送られてきたサービスエージェントSA2000を受け取り、その移動先名2004(C30 SE30)をチェックし、移動先名2004(C30 SE30)に指定されたコンピュータC30に接続されたチャネル(40)を検索し、その結果得られたチャネル(40)を通してサービスエージェントSA2000を送る。

【0288】コンピュータC30上のルーティングマネージャRM30は、送られてきたサービスエージェントSA2000を受け取ると、その移動先名2004の値(C30 SE30)をチェックし、移動先名2004に指定されたサービスエンティティSE30との間に接続されたチャネル400を検索し、その結果得られたチャネル400を通してサービスエンティティSE30にサービスエージェントSA2000を送る。

【0289】サービスエンティティSE30は、チャネル400を通してサービスエージェントSA2000を受け取ると、サービスエンティティSE30の連携インタフェース30がサービスエージェントSA2000の持つ入力データ選択手続き2021を取得し、その手続き2021を実行し、サービスエンティティSE30の持つアプリケーションプログラムAP30に渡すべきデータ項目であるB工程製造速度項目2007を決定する。

【0290】そして、サービスエージェントSA2000の持つ入力データ変換手続き2022を取得し、その手続き2022を実行し、選択された項目であるB工程製造速度2007の値15を取り出し、そのデータがアプリケーションプログラムに渡すデータであることを示すタグを付加した後、サービスエージェントSA2000の持つパラメタ設定手続き2023を取得し、その手続き2023を実行し、アプリケーションプログラムAP30に指定された値15を設定する。

【0291】その後、サービスエージェントSA2000の持つ制御手続き2030を取得し、その手続き2030を実行することによりアプリケーションプログラム

38

AP30の処理を行なう。

【0292】アプリケーションプログラムAP30の処理が終了すると、サービスエージェントSA2000の持つデータ取得手続き2011を取得し、その手続き2011を実行し、アプリケーションプログラムの実行結果を取得する。

【0293】その後、サービスエージェントSA2000の持つ出力データ選択手続き2012を取得し、その手続き2012を実行し、アプリケーションプログラムの実行結果を格納するデータ項目であるB工程所要時間項目2008を選択し、サービスエージェントSA2000の持つ出力データ格納手続き2013を取得し、その手続き2013を実行し、アプリケーションプログラムAP30の実行結果をB工程所要時間項目2008に格納する。

【0294】次に、移動先リスト2002の3番目の要素である(C10 SE10)を取り出し、それを移動先名2004に格納し、チャネル400を通してコンピュータC30上のルーティングマネージャRM30に送る。

【0295】ルーティングマネージャRM30は、サービスエンティティSE30からチャネル400を通して送られてきたサービスエージェントSA2000を受け取ると、その移動先名2004(C10 SE10)をチェックし、移動先名2004(C10 SE10)に指定されたコンピュータC10に接続されたチャネル400を検索するが、コンピュータC30上のルーティングマネージャRM30と接続されているコンピュータの中にはコンピュータ名C10がないため、接続されているコンピュータC20のルーティングマネージャRM20にサービスエージェントSA2000を送る。

【0296】コンピュータC20上のルーティングマネージャRM20は、送られてきたサービスエージェントSA2000を受け取ると、その移動先名2004の値(C10 SE10)をチェックし、移動先名2004に指定されたコンピュータ名C10がコンピュータ名C20と異なるため、サービスエージェントSA2000の移動リスト2003にコンピュータ名C20を格納し、移動リスト2003の値を(C20)とし、サービスエージェントSA2000の移動先名2004(C10 SE10)に指定されたコンピュータ名C10と接続されたチャネル400を検索し、その結果得られたチャネル400にサービスエージェントSA2000を送る。

【0297】ルーティングマネージャRM10は、ルーティングマネージャRM20からチャネル400を通して送られてきたサービスエージェントSA2000を受け取ると、その移動先名2004(C10 SE10)をチェックし、移動先名2004(C10 SE10)とコンピュータ名C10が同一であることをチェックし

39

た後、移動先名2004(C10 SE10)に指定されたサービスエンティティSE10との間に接続されたチャネル400を検索し、その結果得られたチャネル400を通してサービスエンティティSE10にサービスエージェントSA2000を送る。

【0298】このようにして、サービスエージェントSA2000を用いることにより、製造工程A、Bを管理するコンピュータC20、C30に接続された製造機器MA1、MB1を制御するアプリケーションプログラムAP20、AP30を連携して制御することが可能となる。

【0299】また、製造工程が異なる場合でも、製造指示書を作成する際に、その移動先リスト2002に指定する移動先の一覧を変更するだけで制御する製造装置やその順序を変更することができ、容易に連携の方法を変更することができる。

【0300】以上に本発明者によってなされた発明を、前記実施例に基づき具体的に説明したが、本発明は、前記実施例に限定されるものではなく、その要旨を逸脱しない範囲において種々変更可能であることは勿論である。

【0301】

【発明の効果】本願において開示される発明のうち代表的なものによって得られる効果を簡単に説明すれば、下記のとおりである。

【0302】アプリケーションプログラム間の連携処理手続きをサービスエージェントというアプリケーションプログラム本体とは独立した形式で与え、アプリケーションプログラム本体に連携処理のための連携インタフェースを付加し、この連携インタフェースがサービスエージェントの持つ利用手続きを実行することによりアプリケーションプログラム間の連携処理をサービスエージェントを通して容易に変更することができる。

【0303】また、業務に応じて、その業務を進めるために必要なさまざまなアプリケーションプログラムとその操作を組み合わせた利用手順を、サービスエージェントの中に一連の作業手順と各作業で使用するアプリケーションプログラムの利用手続きとして記述し、このサービスエージェントをコンピュータネットワークに接続されたコンピュータ上で動作している各種サービスエンティティの間で移動させ、サービスエージェントを受信したサービスエンティティはサービスエージェントに記述された利用手続きを実行することにより、あらかじめアプリケーションプログラムの連携処理手続きを記述しておかなくても、個々の利用者が自分自身の利用方法に適するアプリケーションプログラム間の連携処理を行えることが可能となる。

【図面の簡単な説明】

【図1】本発明の一実施例である分散システムの構成を示した図である。

40

【図2】本実施例の分散システムのハードウェア構成を示した図である。

【図3】本実施例の分散システムのサービスエージェントの構成を説明するための図である。

【図4】本実施例の分散システムの論理通信路の構成を説明するための図である。

【図5】本実施例の分散システムの論理通信路の構成を説明するための図である。

【図6】本実施例の分散システムの論理通信路の構成を説明するための図である。

【図7】本実施例の分散システムのルーティングマネージャの構成を示した図である。

【図8】本実施例の分散システムの接続先管理テーブルとサービスエンティティ管理テーブルと接続管理テーブルの構成を説明するための図である。

【図9】本実施例の分散システムの連携インタフェースの構成を示した図である。

【図10】本実施例の分散システムのサービスエージェントの条件判定を含む手続きを説明するための図である。

【図11】本実施例の分散システムの入力データ表示手続きの表示例を示した図である。

【図12】本実施例の分散システムの入力データ表示手続きの構造を説明するための図である。

【図13】本実施例の分散システムの操作処理手続き管理テーブルの構成を説明するための図である。

【図14】本実施例の分散システムの関数テーブルの構成を説明するための図である。

【図15】本実施例の分散システムの移動先リストの記述例を示した図である。

【図16】本実施例の分散システムの移動リストの記述例を示した図である。

【図17】本実施例の分散システムのサービスエージェントの移動処理の流れを示すフローチャートである。

【図18】本実施例の分散システムのサービスエージェントの移動処理の流れを示すフローチャートである。

【図19】本実施例の分散システムのサービスエージェントの移動処理の流れを示すフローチャートである。

【図20】本実施例の分散システムのサービスエージェントの移動処理の流れを示すフローチャートである。

【図21】本実施例の分散システムの連携インタフェースのアプリケーションプログラムの実行処理の流れを示すフローチャートである。

【図22】本実施例の分散システムの連携インタフェースのアプリケーションプログラムの実行処理の流れを示すフローチャートである。

【図23】本実施例の分散システムの連携インタフェースのアプリケーションプログラムの実行処理の流れを示すフローチャートである。

【図24】本実施例の分散システムの連携インタフェース

50

41

スのアプリケーションプログラムの実行処理の流れを示すフローチャートである。

【図25】本発明の適用例のコンピュータ統合生産システムの構成を示す図である。

【図26】本発明の適用例のコンピュータ統合生産システムの製造指示書の構造を説明するための図である。

【符号の説明】

C…コンピュータ、AP…アプリケーションプログラム本体、0…ネットワーク、10…サービスエンティティ、11…キーボード、12…マウス、13…入力装置、14…CPU、15…メモリ、16…処理装置、17…表示装置、18…外部記憶装置、19…プリンタ、20…サービスエージェント、30…連携インタフェース、31…データ作成プログラム、32…データ出力プログラム、33…実行制御プログラム、34…データ取得プログラム、35…データ格納プログラム、36…移動先操作プログラム、37…関数制御プログラム、38…アプリケーションインタフェース、40…ルーティングマネージャ、41…管理テーブル作成プログラム、42…接続要求プログラム、43…接続管理プログラム、44…移動管理プログラム、45…接続待ちプログラム、46…接続先管理テーブル、47…接続管理テーブル、48…サービスエンティティ管理テーブル、202…移動先リスト、203…移動リスト、400…チャネル。

【図1】

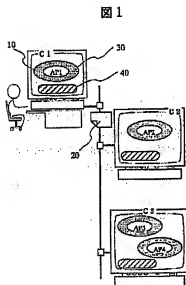
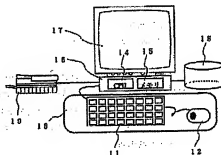


図1

【図2】



【図3】

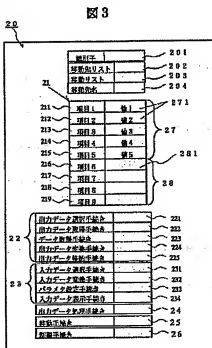


図3

【図4】

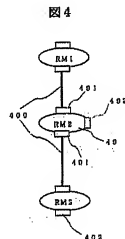
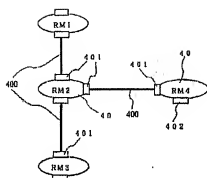


図4

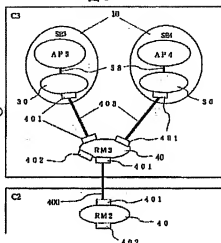
【図5】

図5



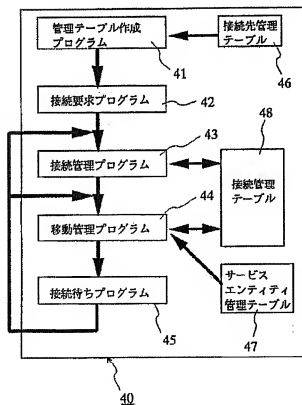
【図6】

図6



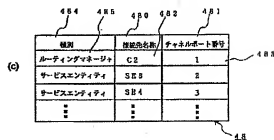
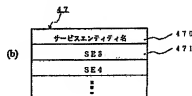
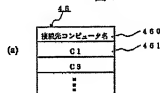
【図7】

図7



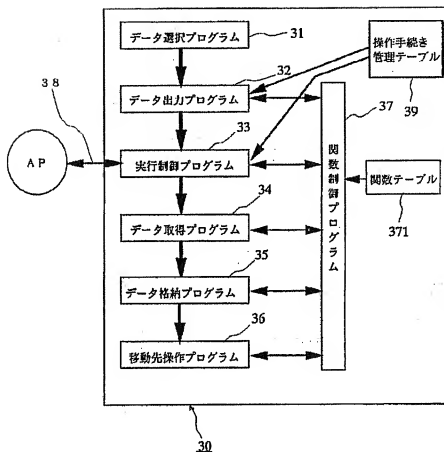
【図8】

図8



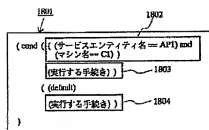
【図9】

図 9



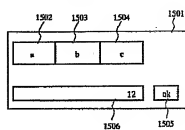
【図10】

図 10

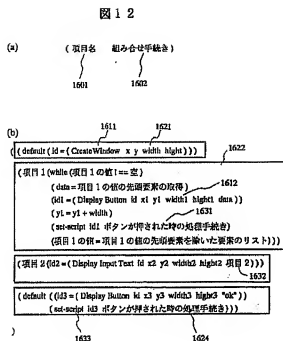


【図11】

図 11



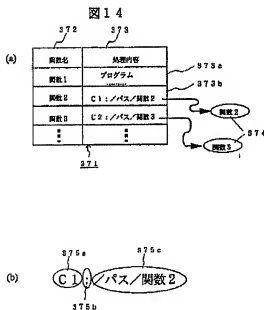
【図12】



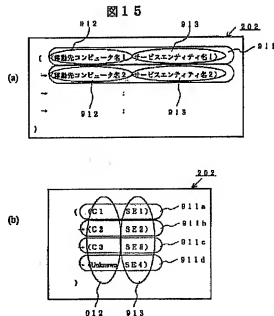
【図13】

| 識別子 | 操作処理手続き |
|-----|---------|
| 3 | プログラム1 |
| 4 | プログラム2 |
| ⋮ | ⋮ |
| ⋮ | ⋮ |

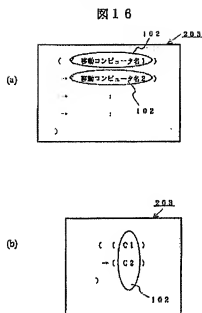
【図14】



【図15】

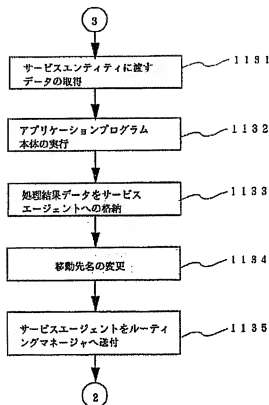


【図16】



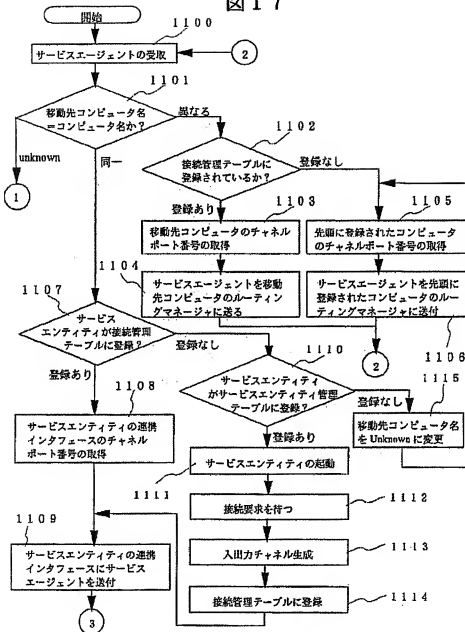
【図20】

図20



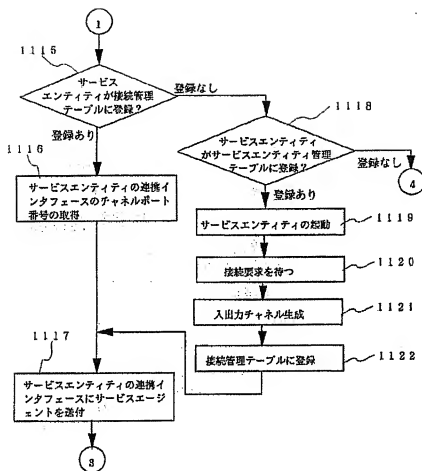
【図17】

図 17



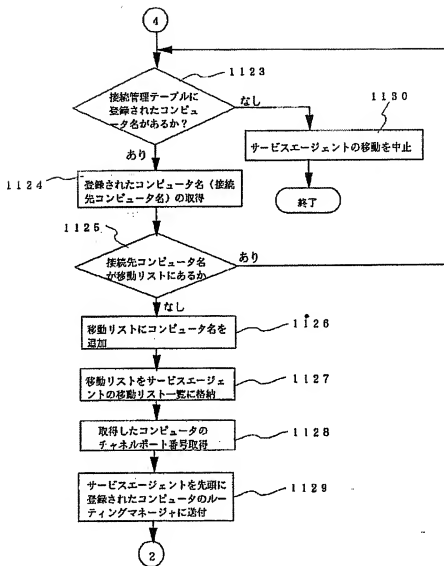
【図18】

図 18



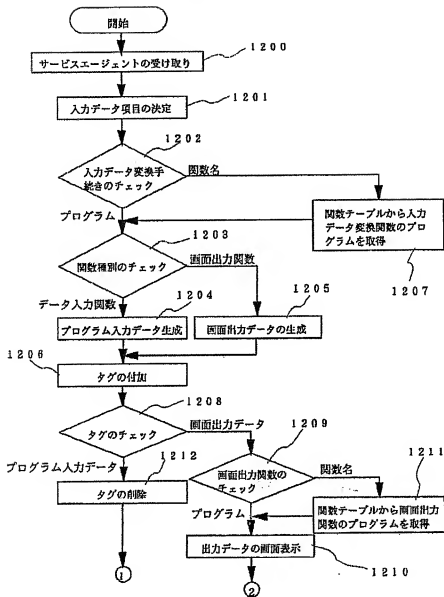
【図19】

図 19



【図21】

図 2 1

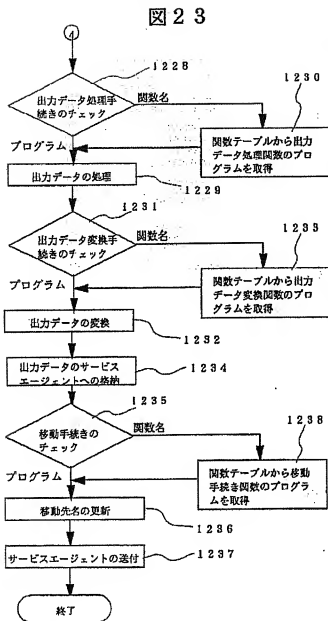


【圖 26】

图 26

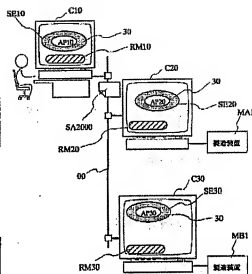


【図23】



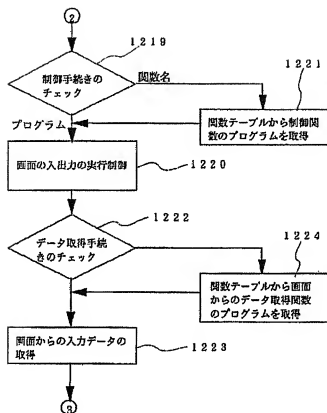
【図25】

図25



【図 24】

図 24



 フロントページの続き

(72)発明者 岡崎 司

宮城県仙台市青葉区一番町二丁目4番1号
日立東北ソフトウェア株式会社内

(72)発明者 青木 靖

宮城県仙台市青葉区一番町二丁目4番1号
日立東北ソフトウェア株式会社内

(72)発明者 中西 潤

神奈川県川崎市幸区鹿島田890番地の12
株式会社日立製作所情報システム事業部内

